1994年11月1日発行（毎月1回1日発行）第13巻11号通巻151号　昭和58年11月2日第三種郵便物認可

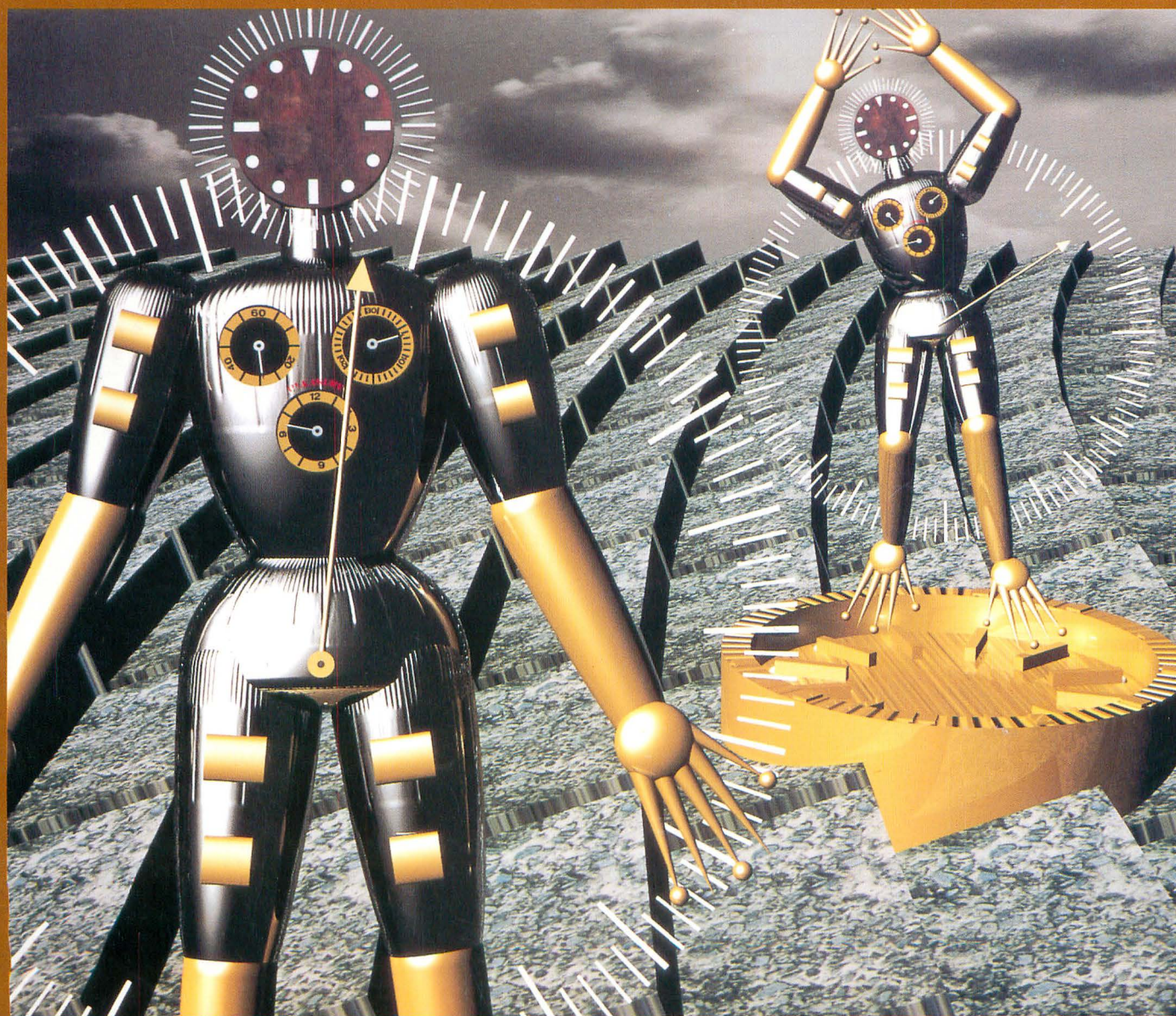
PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

特集　STEP UP BASIC

BASIC修得への道/コーディングの流れを見る/これから始めるSX-BASIC
TGRAPH.FNC/XSPRITE.FNC使用ガイド/外部関数/ライブラリ作成の手引き

MOD.Xのやさしいマニュアル/新製品紹介 BJC-400J

11
1994

SOFT BANK
オー・エックス
定価680円

■実画面：1,024×1,024ドット、表示画：768×：512ドット

●画面は広告用に作成した、機能を説明するためのイメージ画面です。また、各種アイコンなどは、SX-WINDOW ver.3.1がもつ機能を使って作成したもので、標準装備のものとは異なるものもあります。
●本広告中の「シャーペン」で表示している文字のフォントはツァイト社の、「書体倶楽部」のフォントを使用しています。

❶「パターンエディタ」で作成したデータを背景に設定可能。

❷日本語フロントプロセッサ ASK68K ver.3.0の辞書メンテナンスがウィンドウ上で可能。

❸ESC/Page,LIPSIII,PostScriptに対応したプリンタが利用できます。

❹付属アプリケーション「シャーペン」編集例。文字ごとに文字種・文字の大きさの指定、装飾が可能。またインライン入力をサポート、イメージデータの貼りつけもOK。

❺512×512ドットの範囲内で65,536色の表示が可能。

❻「CGAウィンドウ」、65,536色（最大）のコンピュータアニメーション表示が可能。

❼異なる画像フォーマットへのコンバートが可能。

❽アイコンデータや背景データを作成する「パターンエディタ」。

❾オリジナルに作成したアイコンパターンの例。

❿Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。

■「SX-WINDOW ver3.1システムキット」のお問い合わせは…シャープ㈱電子機器事業本部（液映）システム機器推進プロジェクトチーム　〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161（大代表）へ

フィールドが、膨らむ。

先が、ますます面白くなる。

●

未来への確かなビジョンをベースに
発展性のあるプラットホームとしてのウィンドウ環境を提供する
国産オリジナルウィンドウシステムSX-WINDOW。

●

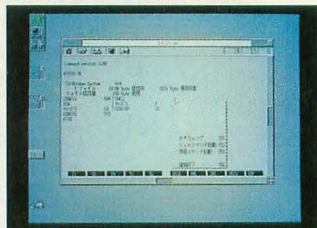
GUI環境や操作環境、高速化へのゆるぎない探求、
マルチメディアの統合的なハンドリング。

●

いま、より多彩なフィールドへ
そのインテリジェンスが展開を始める。

●

次のステージが見えてくる。

●インライン入力のサポート：ASK68K Ver.3.0を利用したインライン入力をSX-WINDOWで実行可能。またシャーペン.Xをワープロとして利用できるよう、さまざまな機能が付加されています。

●コンソールをサポート：Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。
（グラフィックを利用したものなど、SX-WINDOWと処理が重複するものは実行できません。）

●多彩なプリンタに対応：さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ESC/Page、LIPS III、PostScriptに対応したプリンタが利用できます。

今も、先も楽しめる。
いつも新展開の予感、SX-WINDOWのニューバージョン。

SX-WINDOW ver.3.1

「SX-WINDOW ver3.1システムキット」CZ-296SS（130mmFD）/CZ-296SSC（90mmFD） 標準価格22,800円（税別）

EXEディスク2 プレゼント

シャーペンカスタマイズコンテストの力作や、新作SX-WINDOWソフトウェア情報などを満載のディスク情報誌「EXEディスク2」をプレゼントいたします。

●官製ハガキに住所、氏名、EXE会員番号と90mm（3.5型）/120mm（5.25型）の種別を明記の上、お申込み下さい。また、これからEXEクラブへ入会される方は、商品同梱のEXEクラブ入会申込書に「EXEディスク2希望」と明記の上、ご投函下さい。

応募/問合せ先

〒545 大阪市阿倍野区長池町22-22 シャープ株式会社電子機器事業本部システム機器営業部EXEクラブ事務局EXEディスク2係宛
（TEL 06-621-1221大代表）

申込締切

平成6年11月末日消印有効

発送開始

10月20日より順次発送いたします。

●EXEクラブに入っておられない方は、ソフトベンダー「TAKERU」での購入が可能です。
（平成6年11月1日より2ヶ月間、予価200円）

●Z'sSTAFF、書体倶楽部はツァイト社の商標です。●ESC/Pageはセイコーエプソン株式会社の登録商標です。●PostScriptはアドビシステムズ社の登録商標です。●EGWordは株式会社エルゴソフトの登録商標です。

シャープ株式会社

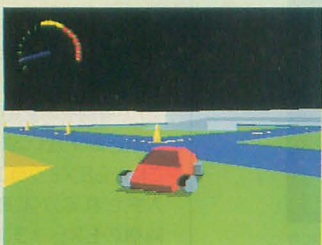
資料請求券 ペリフェラル Oh!X 11係

CONT

〈スタッフ〉

●編集長/前田 徹 ●副編集長/植木章夫 ●編集/山田純二 豊浦史子 高橋恒行 ●協力/有田隆也 中森 章 林 一樹 吉田幸一 華門真人 朝倉祐二 大和 哲 村田敏幸 丹 明彦 三沢和彦 長沢淳博 司馬 護 清瀬栄介 石上達也 柴田 淳 瀧 康史 横内威至 進藤慶到 ●カメラ/杉山和美 ●イラスト/山田晴久 江口響子 高橋哲史 川原由唯 ●アートディレクター/島村勝頼 ●レイアウト/元木昌子 ADGREEN ●校正/グループごじら

特集 STEP UP BASIC

スーパーストリートファイターII

ハードコア3Dエクスタシー

モデリングを楽しんでみよう

BJC-400J

(哲)のショートプロぱーてぃ

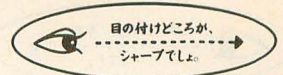
表紙絵：塚田　哲也

E　N　T　S

1994 NOV.
11

UNIXはAT＆T BELL LABORATORIESのOS名です。
Machはカーネギーメロン大学のOS名です。
CP/M, P-CPM, CP/Mupis, CP/M-86, CP/M-68K, CP/M-8000, DR-DOSはデジタルリサーチ
OS/2はIBM
MS-DOS, MS-OS/2, XENIX, MACRO80, MS C, WindowsはMICROSOFT
MSX-DOSはアスキー
OS-9, OS-9/68000, OS-9000, MW CはMICROWARE
UCSD p-systemはカリフォルニア大学理事会
TURBO PASCAL, TURBO C, SIDEKICKはBORLAND INTERNATIONAL
LSI CはLSI JAPAN
HuBASICはハドソンソフト
の商標です。その他，プログラム名，CPU名は一般に各メーカーの登録商標です。本文中では"TM"，"R"マークは明記していません。
本誌に掲載されたプログラムの著作権はプログラム作成者に保留されています。著作権上，PDSと明記されたもの以外，個人で使用するほかの無断複製は禁じられています。

■広告目次

ビデオグラフィックスの
世界へ。
資料請求券
X68030
■お問い合わせは… シャープ株式会社
電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

1,677万色対応、ビデオ映像を高画質・高速取り込み

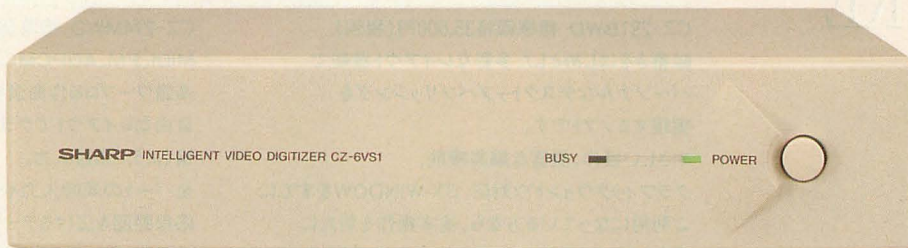
テレビやビデオ、ビデオディスクなどの映像をX68シリーズやMacシリーズ※1の動画・静止画データとして高速取り込みが可能、いわば"ビデオスキャナ"とでも呼びたいビデオ入力ユニットです。1,677万色対応、最大640×480ドットの高解像度※2。動画・静止画の手軽なハンドリングが、新たなグラフィックシーンを創造します。

※1 MacintoshはIIシリーズ以降の機種に対応、ディスプレイ解像度が640×480ドットの場合、取り込み可能な範囲は、160×120ドット、320×240ドットのサイズになります。
※2 X68030/X68000シリーズでは、1,677万色はデータ作成のみに対応。表示は最大65,536色、解像度は512×512ドット。また、Macintoshは機種により表示色数が異なります。

アプリケーションツール「ライブスキャン」を標準装備

動画や静止画を簡単に保存できるアプリケーションソフト「ライブスキャン」※を標準装備。取り込んでいる映像を表示したり、残したいシーンを簡単に静止画保存したり、手軽な動画・静止画ハンドリングでパソコンの可能性をさらに広げます。X68030/X68000シリーズ用SX-WINDOW対応版とMacintoshシリーズ用QuickTime対応版の2種類を同梱しています。

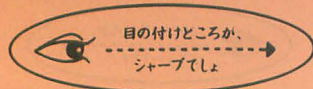
※SX-WINDOW版はバージョン3.0以降（メモリー4MB以上）、QuickTime版はMacintosh漢字Talk7リリース7.1以上のシステムとQuickTime1.5以上（メモリー8MB以上）が必要です。

1,677万色対応の高速映像取り込み、動画・静止画の手軽なハンドリングが、新たなマルチメディアシーンを創造する。

■SCSIインターフェイス採用：パソコンの専用I/Oスロットを使わずに接続可能になり、汎用化を実現しました。またSCSI-2（FAST）インターフェイスの採用により、データ転送速度の高速化を図っています。X68030/X68000シリーズでは、SCSI-2（FAST）対応のハードディスクを接続することにより、パソコン本体を経由しないで、ハードディスクに直接、動画データをテンポラリデータとして記録することが可能です。パソコン本体のハードディスクへは、記録終了後に、テンポラリデータを変換し動画データとして保存できます。

※CZ-600C/601C/611C/602C/612C/652C/662C/603C/613C/653C/663Cに接続する場合は別売のSCSIインターフェイスボードCZ-6BS1ならびにSCSI変換ケーブルCZ-6CS1が必要です。※CZ-604C/623C/634C/644Cに接続する場合は、別売のSCSI変換ケーブルCZ-6CS1が必要です。
※Macintosh Power Bookシリーズに接続する場合は別売のSCSIケーブルなどが必要です。詳しくはMacintosh Power Bookシリーズの取扱説明書をご覧ください。

■高機能MPUを搭載：クロック周波数25MHzの32ビットMPU/MC68EC020を搭載、高速処理やパソコン本体の負担の軽減を実現します。

●MacはMacintoshの略称です。●Macintosh、Macintosh IIは、米国アップルコンピュータ社の登録商標です。●Power Bookは米国アップルコンピュータ社の商標です。●漢字Talk7はアップルコンピュータジャパン社の商標です。●QuickTimeは、米国アップルコンピュータ社の商標です。●価格には、消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は含まれておりません。

標準価格178,000円（税別）

For X68030/X68000series APPLICATION SOFTWARE

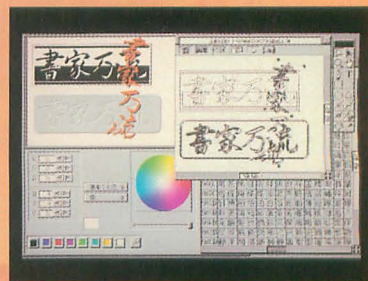
◎独自のアウトラインフォントを付属。フォント&ロゴの作成が自由自在。

フォント&ロゴ デザインツール 書家万流 SX-68K

CZ-282BWD 11月発売予定 NEW

手軽にフォントやロゴが作成できるデザインツールです。
SX明朝体/SXゴシック体フォント(JIS第1水準&第2水準)を付属!

①ベジェ曲線によるアウトライン編集によりフォントデータやロゴデータの作成が可能です。
②フォントファイル全体にわたって傾き/太さ/変形(エンベロープ)のエフェクト処理が可能です。
③既存のフォントファイルからアウトラインデータを抽出しロゴなどを作成したり、ドローオブジェクトに自動的に影をつけるなどのエフェクト処理が可能です。
④複数のフォントファイルをリンクして英数文字、漢字別などにフォント種類を指定したフォントファイルの作成が可能です。
⑤カーニング情報を文字形態から自動的に作成したり、手動で任意の幅に編集できます。
⑥65,536色表示で確認しながらロゴ作成ができるグラフィックウィンドウ(GRW.X)対応。
⑦作成したロゴはクリップボードを介してシャーペン、EGWord SX-68K、XDTP SX-68Kなどのアプリケーションで利用可能です。

※10MB以上の空きのあるハードディスクが必要です。

(4MB、Ver.3.0)

◎パーソナルDTPをX68で

XDTP SX-68K

CZ-291BWD 標準価格35,000円(税別) NEW

縦書きをはじめとした多彩なレイアウト機能でパーソナルなデスクトップパブリッシングを実現するソフトです。
やさしい操作、豊富な編集機能、グラフィックウィンドウ対応、SX-WINDOWをすでにご利用になっている方なら、基本操作を新たに覚えることなく手軽にレイアウトが作成できます。

●豊富なテキスト編集機能 ●65,536色表示に対応 ●多彩な画像フォーマットに対応 ●独立した罫線機能 ●独自のアウトラインフォント(SX明朝体、SXゴシック体の第1水準)を標準添付 ●独立したページウィンドウをサポート ※5MB以上の空きのあるハードディスクが必要です。

(4MB、Ver.3.0)

◎グラフィック感覚の楽譜入力をサポート

MUSIC SX-68K

CZ-274MWD 標準価格38,000円(税別)

MIDI、FM、ADPCMに対応した楽譜ワープロ&作曲演奏ソフトです。
自由なレイアウトでグラフィックを描くように楽譜入力、全パートの同時入力や編集、自動伴奏機能、応用範囲を広げるデータ互換性。
多彩なプリンタ対応で美しい印刷も可能です。

●MIDI、FM、ADPCMを同時に発音、全ての音源を利用した場合、最大発音数は25まで設定可能 ●全パートの同時入力、最大16パートまで編集可能 ●コード&リズムによる自動伴奏機能装備 ●優れたデータ互換性

(4MB、Ver.3.0)

その先のシーンへ。

●さらに実用的なウィンドウシステムへの進化

SX-WINDOW ver3.1 システムキット

CZ-296SS(130㎜FD)/CZ-296SSC(90㎜FD) 標準価格22,800円(税別)

ASK68K Ver3.0を利用したインライン入力のサポート、Human68k/BASICコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できるコンソールのサポートをはじめ、シャーペン.Xをワープロとして利用できるよう機能アップ。また、さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ドローデータ(FSX)/フォントデータ(IFM)処理の高速化も実現しています。

※コンソールでは、SX-WINDOWと処理が重複するものは実行できません。 (4MB)

●定評のGUI対応ウィンドウワープロ

EGWord SX-68K

CZ-271BWD 標準価格59,800円(税別)

ウィンドウワープロとして評価の高いEGWordのSX-WINDOW対応版。キャラクタベースのワープロを超えたグラフィカルユーザーインターフェイス(GUI)による手軽なDTPソフトとしても優れた表現力を発揮します。定評ある日本語入力方式(EGConvert)によるインライン入力、さまざまなグラフィックデータ(GScript)やテキストデータの貼り込み、また文書互換を実現するEDF(Extended Document Format)形式をサポートしています。 (4MB、ver.2.0)

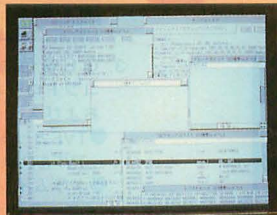
※5MB以上の空きのあるハードディスクが必要です。

●SX-WINDOW開発支援ツール

SX-WINDOW 開発キット Workroom SX-68K

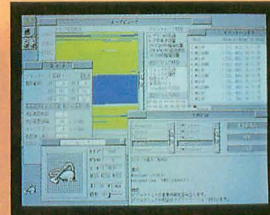
CZ-288LWD 標準価格39,800円(税別)

SX-WINDOW用のソフト開発に必要なツールやサンプルプログラムを装備。プログラムの編集、リソースの作成、コンパイル、デバッグといった一連の作業をSX-WINDOW上で効率よく実行できます。初めてSX-WINDOW用のプログラムに挑戦する人にも、簡単に基本機能の理解が深まる33種(基礎編23種、応用編4種、実用編6種)のサンプルプログラム付き。

※ご使用に当ってはC compiler PRO-68K ver.2.1が必要です。 (4MB、ver.2.0)

●SX-WINDOW開発キットのサポートツール

開発キット用ツール集

CZ-289TWD 標準価格12,800円(税別)

SX-WINDOW開発キットをさらに使いやすくするためのツールです。SXコールの簡易リファレンスを簡単に検索するインサイドSX、イベントの発生を常時監視・確認するイベントハンドラ、リアルタイムにメモリブロックの利用状況を表示するヒープビューアなど11種のツールが用意されています。 (4MB、ver.2.0)

●SX-WINDOW対応ドローイングツール

Easydraw SX-68K

CZ-264GWD 標準価格19,800円(税別)

イラスト、フローチャート、地図、見取り図など各種グラフィックが製図感覚で作成できます。作成したデータは他のSX-WINDOW対応アプリケーションでも利用でき、企画書などの作成をサポート。ページプリンタドライバも標準装備。 (4MB、ver.3.0)

●ウィンドウ対応グラフィックツール

Easypaint SX-68K

CZ-263GWD 標準価格12,800円(税別)

マウスによる簡単操作、65,536色中16色の多彩な表現、クリエイティブマインドに応えるウィンドウ対応ペイントツールです。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間でのデータ交換もできます。 (2MB、ver.1.1)

●SX-WINDOWを楽しく使うためのアクセサリ集

SX-WINDOW デスクアクセサリ集

CZ-290TWD 標準価格14,800円(税別)

SX-WINDOWをさらに便利に楽しく使うためのデスクアクセサリ集です。スクリーンセーバ、スクラップブック、スケジューラ、アドレス帳、電子手帳通信ツール、パズルなど、12種の豊富なアクセサリが収められています。 (4MB、Ver.3.0)

●マルチタスク機能をはじめ通信環境がさらに充実

Communication SX-68K

CZ-272CWD 標準価格19,800円(税別)

通信環境をさらに高めたウィンドウ対応の通信ソフトです。マルチタスク機能により他のアプリケーションを実行中でも簡単に通信が可能。自動ログイン機能やプログラム機能、など豊富な機能をサポートしています。 (2MB、ver.1.1)

●FM音源サウンドエディタ

SOUND SX-68K

CZ-275MWD 標準価格15,800円(税別)

他のミュージックソフトで演奏中の音色を、簡単に作成、変更できるマルチタスク機能、またエディット、イメージ、ウェーブの3つの編集/確認モードを装備。作成中の音色も50曲の自動演奏でリアルタイムに確認、編集できます。 (2MB、ver.1.1)

●SX-WINDOW対応になってさらにパワーアップ

倉庫番リベンジ SX-68K ユーザー逆襲編

CZ-293AW(130mmFD)/CZ-293AWC(90mmFD) 各標準価格6,800円(税別)

倉庫番10年にわたるユーザーの投稿など、新作306面が目白押し。まさに倉庫番の最強版がSX-WINDOW上で楽しめます。AI機能やエディット機能、キャラクタ変更機能も装備。半年で解けたらあなたは天才?です。 (2MB、ver.1.1)

PRO-68K シリーズ

●X68030/X68000対応

CZ-295LSD 標準価格44,800円(税別)

※メインメモリ2MB以上が必要です。

C compiler PRO-68KのX68030/X68000対応版。MPU68030、MC68882の命令セットに対応したアセンブラ、デバッガ、ソースコードデバッガを付属。またHuman68k ver.3.0、ASK68K ver.3.0にも対応。新たにGPIBライブラリ、MC68882対応フロートライブラリを付属しています。

※ (2MB,ver.1.1) の表示は、メインメモリ2MB以上、SX-WINDOW ver.1.1以上が必要であることを示します。

※発売予定のソフトの画面は実物とは異なる場合があります。

●EGWord、EGConvertは株式会社エルゴソフトの登録商標です。

●お問い合わせは…シャープ㈱電子機器事業本部(液映)システム機器推進プロジェクトチーム 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ シャープ株式会社

資料請求券 CZソフト Oh!X 11月

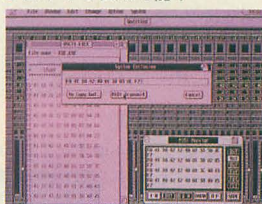
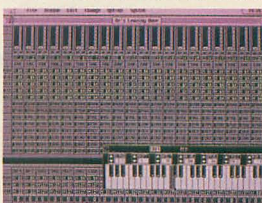
SOUND Canvas GS音源対応
MIDIマルチレコーダー

Mu-1 Musicstudio GS
[ミューワンジーエス]

Mu−1GSはローランド社SC−55mkⅡなどGS音源をフルに活用できるコントロール群と高度な音楽表現を可能にする新感覚エディットウインドウ搭載のMusicstudioプロフェッショナルバージョンです。

特長

1. 新感覚エディットウインドウ
   - エディタ感覚のプロフェッショナル仕様

2. 簡単エクスクルーシブ入力
   - チェックサム自動計算入力
   - 曲中でも使用可能！

3. GS音色コントロール機能
   コントロールコードのリアルタイムコントロールおよびステップ入力が可能
   - TVFカットオフフリケンシー、TVFレゾナンス、TVF&TVA・アタック、ディケイ、リリース・タイム
   - ドラムインストゥルメント・ピッチ、リバーブセンドコーラスセンド、パンポット、ボリューム

4. RCPコンバート機能追加
   - カモンミュージックRCM、STED2
5. 24トラック／リアルタイム録音／ステップ入力機能
6. X68030(25MHz)／Human68K Ver.3.01対応
7. RS−232C／MIDI出力対応
   (注意：出力のみ対応、単独使用不可／要MIDIボード)
8. 内蔵FM／ADPCM音源対応
9. 国本佳宏／GS対応デモ曲収録

データコンバート一覧表

読み込み (Load)		ファイル
Mu−1GS←	ミュージ郎／ミュージ郎Ⅱ MUSIC PRO-68K FM&MIDI MML MML スタンダードMIDIファイル フォーマット0/1 RCM／STED2	SNG MUS OPM MID RCP
書き込み (Save)		**ファイル**
Mu−1GS→	スタンダードMIDIファイル フォーマット1 MUSIC PRO-68K FM&MIDI	MID SCO

ハード構成 シャープ68000／030本体
MIDIボード（シャープ社製CZ-6BM1またはシステムサコム社製SX-68M／SX68MⅡ）
ローランド社製GS対応音源SC-55、SC-55mkⅡ SC-300、SC-500など

Mu-1 GS 標準価格 ¥28,000（税別）

■本ソフト動作には、メインメモリ2MBが必要です。

スタンダードMIDIファイル オリジナルアーティストシリーズ 各¥3,500

好評発売中！

SCB-1001

SCB-1002

SCB-1003

SCB-1004

SCB-1001 duplicity／佐久間正英
SCB-1002 ブレインボックス美術館／国本佳宏
SCB-1003 PICES OF WORKⅡ／本多俊之
SCB-1004 HOPE／松居慶子

■推奨音源：Roland SC-55, SC-55mkⅡ, SC-88
SC-33, CM-500, CM-300

■「GS対応エクスクルーシブデータ」を使用しています。GM音源など推奨音源以外の機器を使用する場合、音量等のバランスが異なりますのでエディットしてお聴きください。

■SC-88対応「スタンダードMIDIファイル クラシックシリーズ」6タイトル発売予定

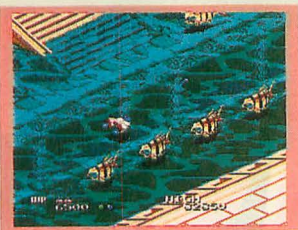
〈星智輝 with T. T. CDシングル〉
「No Frame No Fame/Christmasday Ring Our Hearts」
定価 ¥1,000（税込）
☆九十九電機パソコン本店Ⅱ4Fで販売しております。どうぞご利用下さい。

〒213 神奈川県川崎市高津区下作延1043
TEL 044−855−4335

VIEWPOINT™

ビューポイント

クォータービューの魅力炸裂

11月、発進!

※画面は開発中のものです。

シューティングの名作、いよいよX68000版で登場

クォータービューによる3D表示、バイオメカニカルな敵キャラクターの滑らかな動きでゲームフリークスを熱狂させた「VIEW POINT」。アーケードゲームで大人気を博した「VIEW POINT」が、いよいよX68000版で登場。X68000シリーズならではのハードウエア特質を活かし、アーケード版のイメージを損なわない操作性とダイナミックなグラフィック画面を実現。FM内蔵音源をフル活用、サウンド面においてもアーケード版のクオリティを忠実に再現。「コンフィグ画面」を新たに設定し、難易度の変更も自在。ビギナーからシューティングマニアまでをカバーする4段階の難易度が魅力。

X68000シリーズ

- X68000／30シリーズおよびXVI対応 ●2HD2枚組(5")
- 要2MB ●ハードディスク対応 ●内蔵FM音源対応
- キーボードおよびジョイスティック操作対応

予定価格7,800円（税別） ●11月発売予定

- 全国のパソコンショップにてお買い求めください。
- 通信販売（送料無料）をご希望の方は、住所・氏名・郵便番号・商品名・機種名・掲載メディア名を明記し、現金書留にてお申し込みください。
- 本ソフトを当社に無断で複製すること、及び賃貸（レンタル）、中古販売（疑似レンタル）については、これを一切許可しておりません。

知的世界とあなたをむすぶ

グラフィックデザイナー等のゲーム開発者を募集しております。
詳しくは、TEL.03-5474-3581代までご連絡ください。

株式会社 ネクサスインターラクト
〒106 東京都港区六本木7-21-7 ウェスタ六本木 TEL.03-5474-3581代
●ゲームソフトの内容に関するお問い合わせは TEL.03-5474-3584/ユーザーサポート係

パーソナルコンピュータ・ゲームマガジン
ゲームブラスト

GAME BLAST

毎月8日発売
定価530円（税込）
お近くの書店で
ご予約・お買い求めください

10月8日創刊

特集 1
パソコンゲーマーだけに贈る
マルチメディア実地検証

特集 2
IBM PC用ジョイスティックをPC-98シリーズで使い倒す！

海外ロケ最新作
ブルース・シカゴ・ブルース

特報！
ブランディッシュ3

RPG復活宣言
ルナティックドーン II

創刊記念特別付録 3.5"体験ディスク（PC-9821専用）
3Dロボット格闘ゲーム
メガトンアームズ

ⓒ1991 by Reactor Inc. "Spaceship Warlock" is a trademark of Reactor Inc. All rights reserved.

欲しいパソコンが当たる！ 当たる!!

Hello! PC, GAME BLAST 連続創刊記念パソコンプレゼント！

クイズに答えてご応募下さい。正解者の中から抽選で下記の賞品をプレゼントします。

【賞品】
- ■ PRESARIO（COMPAQ）……2名
- ■ WOODY（Panasonic）……2名
- ■ UpGrade-MULTI（エプソン）……2名
- ■ FM TOWNS II Fresh（富士通）……2名
- ■ PS/V Vision（日本IBM）……2名
- ■ Macintosh LC575（アップルコンピュータ）……2名
- ■ PC-98 MULTi（NEC）……2名

また、応募者全員の中から200名様に"オリジナルテレホンカード"をプレゼント！

【問題】○の中に当てはまる文字は何でしょう？ ①〜③の中から選んで下さい。

ソフトバンクが9月と10月に創刊する雑誌名は

Hello!○CとGAME○LAST

①PとB ②AとZ ③XとY

【応募方法】

官製ハガキに、クイズの答え、ご希望の賞品名1つ、住所、郵便番号、氏名（フリガナ明記）、電話番号、年齢、性別、職業を明記の上、下記までご応募下さい。●応募先 〒103 東京日本橋郵便局留 ソフトバンク「創刊記念パソコンプレゼント」係 ●締切 平成6年11月8日（火）必着 ●抽選/発表 厳正なる抽選により当選者ご本人に直接通知させていただきます。※当選後の権利の譲渡・商品の交換・換金には応じられません

ソフトバンク株式会社/出版事業部
〒103 東京都中央区日本橋浜町3-42-3 TEL.03-5642-8100（販売）
SOFT BANK

初心者にやさしいパソコン情報誌〈ハロー！ピーシー〉

Hello! PC

11月号［創刊第2号］ 10月8日発売! 定価390円 毎月8日発売

［第1特集］
ここまでできる活用術
ノート、サブノートのこれが正解!
話題の製品を一堂に集め、役に立つ活用法と新たな使い方を提案し、サブノート、ノートパソコンの魅力に迫ります。

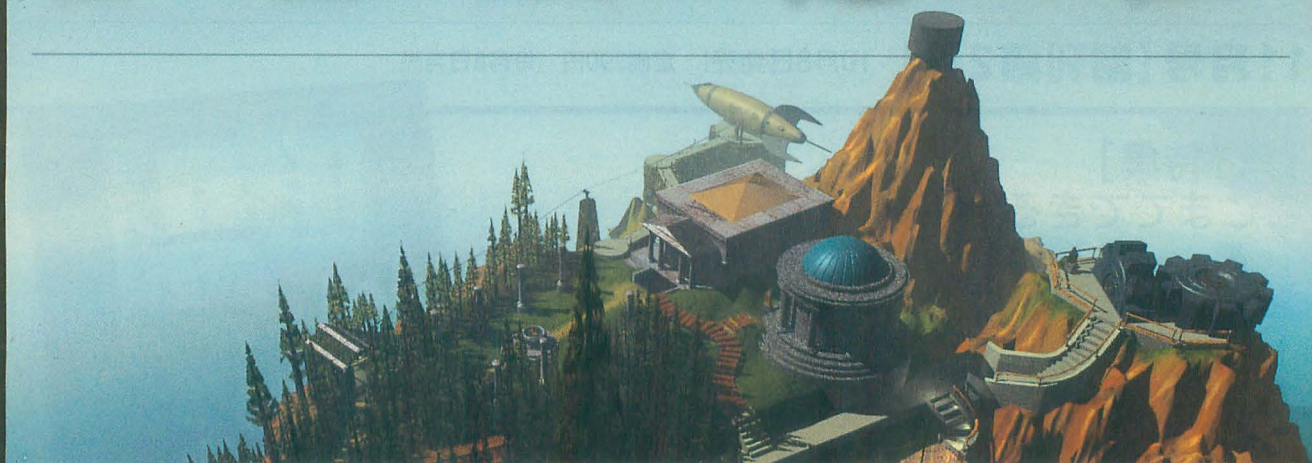
［第2特集］
深くおつきあいするために、知っておきたい
Windowsの謎
知れば知るほど楽しく使えるWindows。その基礎からちょっと得するテクニックをわかりやすく解説します。

［速報リポート］
各メーカーから新製品も登場!?
データショウ'94

特別付録 40Pオールカラー
周辺機器がわかる本①
「ディスプレイ編」
全28機種の詳細データに加え、専有スペースが一目でわかるスケールデータも掲載したディスプレイカタログ。初めての人でもわかる用語辞典付き。

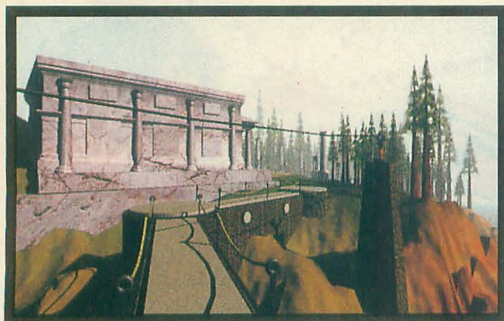
入門講座
●ウィンドウズ ゴー！ゴー！
●めざせ合格 漢字Talk7 予備校

SOFT BANK ソフトバンク株式会社／出版事業部
〒103 東京都中央区日本橋浜町3-42-3
TEL 03-5642-8100

MYST™

バーチャル感覚の次世代
アドベンチャーゲーム登場!

図書館、プラネタリウム、ロケット、時計台、巨大歯車
舟の模型が沈んだ池が点在するMYST島。

隠された謎を求めて、プレイヤーは、一冊の本を手がかりに島に
点在する数々の謎と仕掛けを、パズルを解くように解明していく。

1994年12月発売予定
SONY PlayStation版
発売元ソフトバンク
予価8,800円

深い霧の中で待ち受ける展開、予想外の出来事、
仮想世界MYST島と4つのミステリアスな
幻想の世界を冒険する
新感覚のインタラクティブムービー型
アドベンチャーゲーム。

11月29日
ソニー次世代機PlayStation専門誌
THE PLAYSTATION
創刊
話題の新作情報満載!!

"PlayStaion"
は株式会社ソニー・コンピュータテインメントの商標です。
Software Copyright1994.Cyan,Inc.and Sunsoft. All right reserved.

SOFT BANK ソフトバンク株式会社
東京都中央区日本橋浜町3-42-3 TEL.03-5642-8145

SEGA SATURN MAGAZINE
[セガサターンマガジン]

次・世・代
ゲーム雑誌も
NEXT GENERATION!!

セガ次世代
ゲームマシン
セガサターンの情報を
パーフェクト・サポート!

11月下旬
創刊!

SEGA SATURN

SOFT BANK ソフトバンク株式会社/出版事業部

©1994 SEGA

新製品紹介

BJC-400J

Taki Yasushi 瀧 康史

キヤノンのプリンタBJシリーズの新モデルが発売されました。カラープリンタで低価格の注目機です。9月号で紹介したエプソンのMJ-700V2Cの強力なライバルとなりそうなこのBJC-400J，機能を比較しながら見てみましょう。

69,800の低価格

キヤノンから今回発売されたBJC-400J。お値段(定価)は69,800円なり。正直これは安い。このお値段で，360dpiカラーバブルジェットプリンタときている。しかも，モノクロで使ったときには速い速い。コード印字が好きになっちゃいそうだ。

BJC-600J(10万円以上した同社の360dpiカラーバブルジェットプリンタ)とBJ-220JCII(69,800円のモノクロバブルジェットプリンタ)を買った人は，悔しくて夜も寝られんって感じ。

モノクロ時のみスムージング機能を使い，720dpiが使えるだけで，普段は360dpi。先に発売されたエプソンのMJ-700V2Cよりは，dpiが劣ってしまうが，あれが定価99,800円だったことを考えると，なかなかいけそうな気がする製品だ。

X680x0との相性

もうばっちり。

と，ひと言で終わってしまうが，これで済ませては仕方がないので，どうばっちりなのか書いてみよう。

BJシリーズといえば，48ドット(360dpi)の低価格プリンタとして発売されてからもう長い。だから，SXシステムにも間違いなく対応している。BJ-＊＊＊＊＊の設定で，モノクロ印刷もカラー印刷もばっちり。しかもモノクロ印刷の処理速度は，結構速いときている。

印字例の，馬鹿みたいに装飾をつけたものを見てわかるとおり，相性についてはまったく問題ないといってよいと思われる。

BJC-400J　69,800円(税別)/キヤノン販売　☎03(3455)9544

まあ，SXならば同じdpiならどのプリンタで印刷しても同じ結果なので，この点は心配はない。次にそれ以外での印刷についてみてみよう。

MATIERにも対応

BJシリーズは基本的にどの機種も制御コードが同じなので，MATIERでも問題なく使うことができる。設定は「BJC-820J」にすればいい。ただし，BJC-820JはA3プリンタなので，BJC-400Jで横幅最大に印刷するときには，「最大サイズ」ではなく，自由サイズで200mmで設定しなくてはならない。それでも，ややこしい設定などは何もせずに従来のソフトが使える点はおいしい。

買ってきてすぐに使えるというのは，やっぱりうれしいものだ(特に最近はハードを買ってきても，対応ソフトを作ってから使うことが多かったからかもしれないが)。MATIERの印刷は，クオリティもなかなか高いし，グラフィックをロードしてすぐに印刷できるので，なかなかお手軽だ。思わず楽しくなって，イメージスキャナで取り込んだ，デニス・ニューマン(私の好きなボディビルダーのひとり)の写真を貼りまくってしまった。

dpiが劣る分，同じ画像の印刷ではMJ-700V2Cよりはキメの細かさで劣るが，それでも360dpiの印刷はなかなか綺麗である。

TeXでの印刷

X680x0ユーザーには，TeXという言葉は，すでに大きな地位を占めていると思う。

TeXは，環境さえセットすれば，基本的にたいていのプリンタで問題なく利用できるので，BJC-400Jも当然ながら，BJシリーズの環境設定で使うことができる。

今回使用した環境は，BJ-220JCIIのものである。ただ，先月，丹氏が作成した環境

ファイルは，なぜかA4ファイルの印刷がうまくいかなかったので多少手直しした。リスト1を入力して環境ファイルに使えば，そのままBJ-220JCIIと同じように使えるはずである。動作確認はあまり行っていないのだが，今月のTeX入門のページに掲載したサンプルもこのBJC-400Jで印刷したものなので，たぶん大丈夫だろう。

なお，詳しくは，TeX本でも読んで環境設定をしてほしい。

スムージング機能

スムージング機能とはどういうものかというと，360dpiで送られたデータのジャギー(ギザギザ)を，720dpiという倍のdpiで埋めるという機能だ。ということは，送ったデータをうまく加工してくれて，通常以上に綺麗な出力ができるはずである。

スムージングONにするには，ディップスイッチで設定をするだけでなく，白黒専用カードリッジを使わなくてはならない。

とりあえず，シャーペンで印刷してみたが，白黒専用インクを使ったために，色が出ず，黄色は出力されていない。これはシャーペンの仕様だ。

見ればわかるとおり，スムージングされて滑らかになっているが，全体的に文字が太くなってしまっている。半角文字には如実に違いが現れている。ただし，この出力結果が好みかどうかという問題があるし，スムージングモードを使うと処理が遅くなるという欠点もあるから，場合によってはあまり楽しい機能ではないかもしれない。私には文字が滲んで見えるし，何よりも，

今回発売されたBJC-400J。
800円なり。正直これは
段で、360dpiカラーバブ
タときているからね。し
使った時には速い速い。コ
なっちゃいそうだ。
上した同社の360dpiカ
トプリンタ)とBJ-220JCII
ロバブルジェットプリン
悔しくて夜も寝られん

シャーペン(ノーマル)でのカラー印字(専用紙)

印刷が妙に遅くなってしまうのが嫌だ。

なお，この実験をしてみてわかったのだが，白黒専用インクを使って，スムージングをOFFにすると，カラーインクを使うよりも印刷が格段に速い。TeXなどは白黒なので，こういう印刷をするときには，わざわざ白黒インクに取り替えるのもいいかもしれない。

その他

MJ-700V2Cでは，「専用紙」と「普通紙」とで，印刷結果の違いが大きい。たしかに，専用紙で印刷するととても綺麗なのだが，普通紙に印刷すると，やはりところどころ，髭が出てしまう。

カラーグラフィックを印刷する場合，だいたいにおいて，できるだけ綺麗に印刷しようと専用紙を使うだろう。だけど，白黒で印刷するときには，普通紙を使うことのほうが多いと思われる。この点は，BJシリーズが新機種の登場のたびに発展してきたところだ。たとえば，同じBJシリーズで

今回発売されたBJC-400J。
,800円なり。正直これは
段で、360dpi　ラーバブ
タときているからね。し
使った時には速い速い。コ
なっちゃいそうだ。
以上した同社の360dpiカラ
プリンタ)とBJ-220JCII(69,
バブルジェットプリンタ)を
くて夜も寝られんって

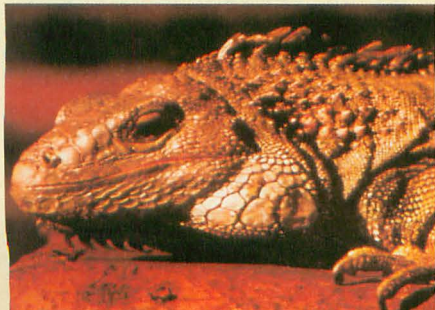
スムージングONでの印字(専用紙)

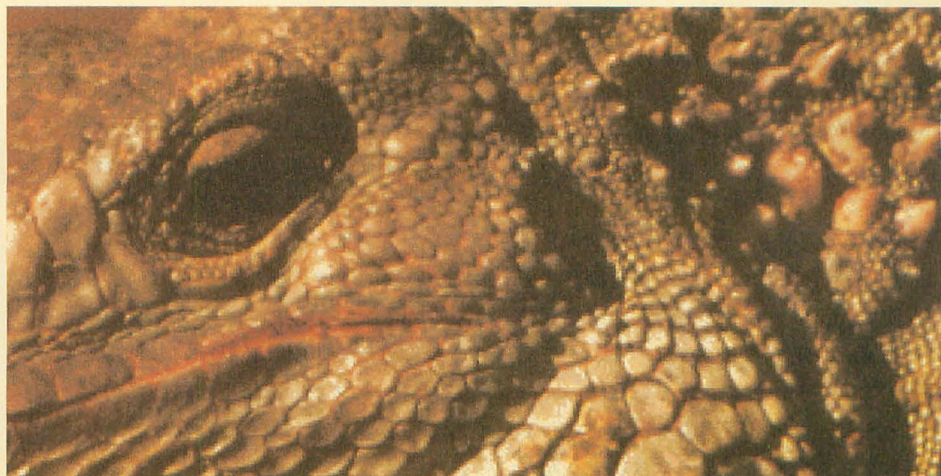
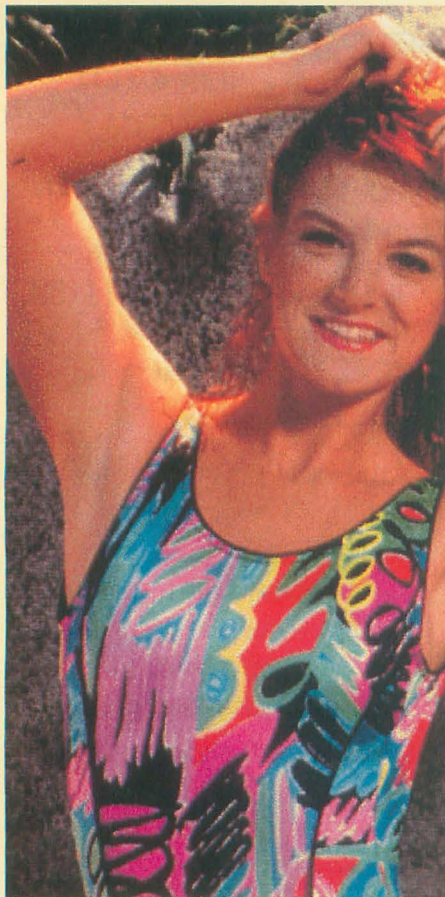
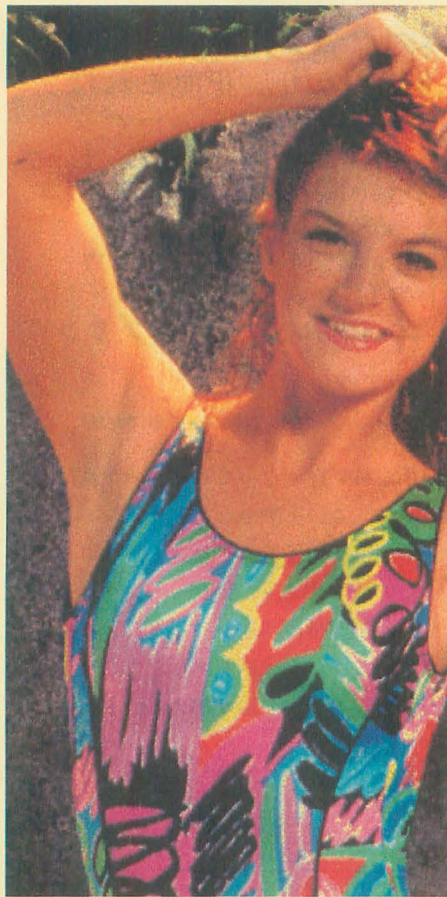
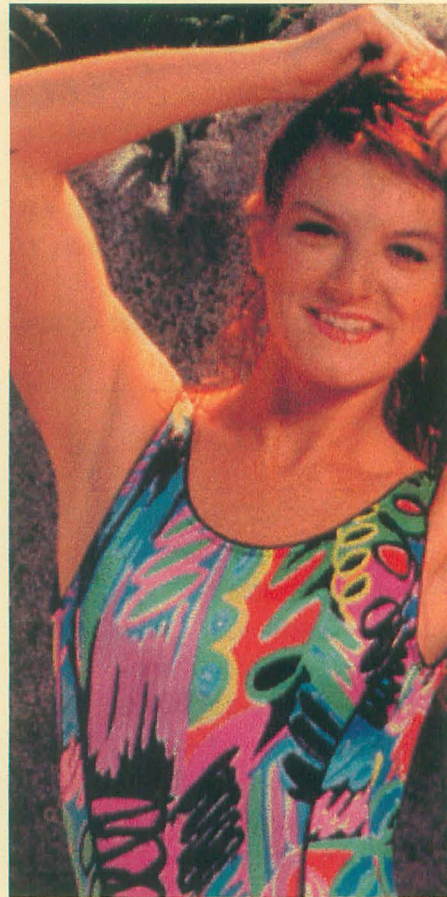
も，BJC-880JというLIPS VIcを搭載した40万円以上するバブルジェットページプリンタが同社から出ているが，こちらの印刷よりも，今回発売されたBJC-400Jのほうが，普通紙への印刷は綺麗だと私は感じている。たしかにBJC-880Jは上位のモデルだが，基本的な性能に関していえば，BJシ

リスト1　コンフィグレーションファイルの例

```
-remark=CANNON_BJ220JC_Config
-dpi=360
-remark=-TRAM
-remark=-GRAM
-width=2880
-remark=-width=2816
-height=3960
-remark=-height=4032
-remark=-xOffset=-90
-remark=-yOffset=-180
-pinBytes=8
-MSBisUpper
-init=¥e@¥e3¥x20¥er%0
-remark=-init=¥e@¥e3¥x20¥er%0
-CRLF=¥n
-remark=-CRLF=¥r¥n
-FF=¥f
-graphic=¥e|B¥x40%2I
-start=
-relative=¥e¥¥%2/2i
-remark=-start=¥e$%2/6i
-remark=-relative=
-repeat=
-dump=LPT
```

MATIERでの印刷

画面の色により近い印刷を実現するために，
青を強調して色調整を施している

▲画面写真

▶BJC-400J(MJ用スーパーファイン専用紙)

MJ-700V2C(スーパーファイン専用紙)

BJC-400J(BJ専用紙)

BJC-400J(MJ用スーパーファイン専用紙)

◀画面写真

Windowsによる印刷例である。スーパーファイン専用紙とは，エプソンから発売されているカラー印刷専用紙で，2,400円/200枚とやや高価だが高品質のもの。ここに挙げた印刷サンプルのなかでは，スーパーファイン専用紙にMJ-700V2Cで印刷したものが画面で見た色にいちばん近かった。BJC-400Jでの印刷も，このスーパーファイン専用紙を使用したほうが発色がよい。美しい印刷を望むのなら紙の選択は重要なポイントであるようだ。なお，今回は，印刷時間や，紙，インクのランニングコストについては検証していない。

リーズは新製品が出るたびに発展しているのだ。

プリンタの設定のほうは，X680x0でもまったく問題はない。ディップスイッチもちゃんとあるので，コード印字のフォントの¥マークをバックスラッシュにすることもできるはずだ。BJ-220JCIIと違ってLEDパネルはないが，これは大した問題ではないだろう。

それから，何度もいうように，今回のBJC-400Jは，MJ-700V2Cに比べて定価が3万円も安い。dpiは半分なので最大印字品質には結構違いが出てくるのだが，そこまで高品質でなくともかまわないというユーザーによっては，この3万円の差は大きいかもしれない。実売価格でどのくらい違うかが勝負かも。360dpiでも結構楽しめるものだし，なんといってもお手軽だ。

MJ-700V2Cが本当に高品質のカラー印刷を楽しみたい人をターゲットにしているのに比べて，BJC-400Jは，品質はやや劣るもののほぼ満足できるカラー印刷で価格のほうを抑えてあるといえるだろう。いままでモノクロで十分，と思っていた人たちも，プリンタ購入の際の選択肢のひとつに加えてもよいと思われる価格設定である。

カラーインクの価格もひと昔前より，ずっと安くなってきた。ワープロしか使わないからカラー印刷は必要ないと思っている人も，グラフィック印刷の誘惑にかられてみるのも悪くないかもしれない。

MJ-700V2Cの印字品質について

MJ-700V2Cの名誉のためにいっておかなくてはならないことが2，3あります。

9月号と10月号にMJ-700V2Cでの印字サンプルを掲載しましたが，あれには少し問題点がありましたので，ここでちょっと補足します。

まず，レビュー記事を書いた当時は，品薄状態で専用紙が手に入らず，印字サンプルもほとんどが普通紙を使っています。

それから，グラフィックの印刷例ですが，これはグラフィックのほうで最初から2分の1に縮小したものでした。画面の情報量の違いは，印刷品質にかなり影響してしまいますので，掲載したサンプルは，どこまで綺麗に印刷できるかの限界を示したものではありません。付録ディスクのhcescv2xでは，同じサイズでドットを増やして印刷できます。

比較のために条件を同じくするために，あのような形での掲載になったというわけです。

NEW PRODUCTS

新製品紹介

Free Software Selection Vol.2

Taki Yasushi 瀧 康史

Free Software Selection Vol.2　6,000円（税別）／計測技研　☎0286(22)9811

待望のフリーセレクション

X68000ユーザーの環境がフリーソフトウェアで成り立っている場合が多いのは事実である。私の環境にも，多くのフリーソフトウェアが使われている。編集部の環境も然り。優れた市販ソフトが少ないという逆境のもと，昔からX68000の世界ではよくできたフリーソフトウェアが出回っていた。

今回発売された「Free Software Selection Vol.2」（以下フリセレ2）は，そういうフリーソフトの集大成だ。ほとんどはパソコン通信を通じて入手できるものだが，なかには，特定のBBSへアクセスしなければ手に入りにくいものもある。NetBSD（いわゆるUNIX）をはじめ，mule（いわゆるNEMACSの多国語版）など，大きすぎてダウンするのが困難なものもある。そういう意味でなかなか有意義なCD-ROMだ。

さて，私は「Vol.1」のほうを見ていないので，それとの比較はできないが，フリセレ2はデータ集としてみれば，収録数はそれほど多くはない。基本的にネット上での投稿による募集だったためか，グラフィックや音楽データはあまりないようだ。フリーソフトというからには，オリジナルでなければならないだろうしね。

しかしながら，ツール集としてみれば，なかなかポイントが押さえられている。有名どころの，Oh!Xでも名前が出てくるようなツールはたいてい収録されているようだ。

いずれにしても，通信をしていないユーザーには特にお買い得かもしれない。

ソフトの楽しみ方

このソフトを楽しむには，SCSI-IIのCD-ROMドライブと「CD-ROM Driver ver.2.00」が必要だ。CD-ROMドライブはSCSI-IIに対応していればどれでも使えるはず。しかし，「CD-ROM Driver ver.2.00」のオーディオCDへのアクセスは，東芝，ソニー，PLEXTOR，松下寿製のドライブで動作確認はされているが，保証されているわけではないようだ。詳しくはコラムを読んでほしい。

ドライバ（cddev.sys）を購入したら，あとはただ，CONFIG.SYSに，

DEVICE＝¥パス¥CDDEV.SYS

の1行を挿入すれば，CDDEV.SYSがIDの順にデバイス*1を勝手にサーチし，CD-ROMにドライブ名を割り当てる。サーチには多少時間がかかるので，それが嫌ならば/id6というようにオプションをつければよい（ここでの6がCD-ROMのID）。

こうやって起動すれば，ISO9660のCD-ROMが読めるようになる。フリセレ2もISO9660フォーマットなので，これで大丈夫。もちろん，SX-WINDOWからもCOMMAND.Xからもアクセスできる。ただ，一部のファイルメンテナンスソフトウェアではドライブとして認識されないので注意が必要だ。余談だが，MS-Windows用CD-ROMやMEGA-CDのCD-ROMもISO9660だから，これらのソフトが身近にあれば，覗いてみると楽しいかもしれない。

それから，Human68kはver.2.03以降が使えるが，できるだけver.3.00以降を利用すること。最新バージョンは「SX-WINDOW ver.3.1」に添付のver.3.02なので，購入しておいたほうがなにかと便利だろう。

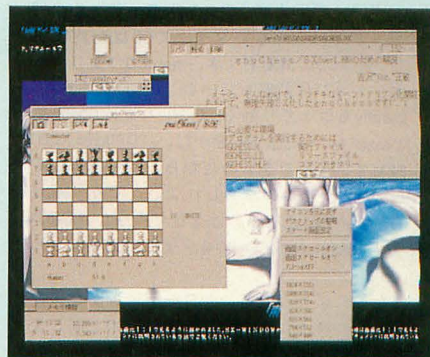
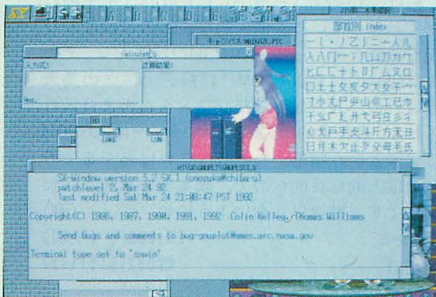
＊1――その機器が何物であるか，たとえばここではCD-ROMデバイスを探す。

CD-ROMドライブユーザーへ

では，フリセレ2の中身を紹介しよう。

最初はCD-ROMドライブを有効活用するのに便利なものから始めよう。

まずはCDROM.XとCD2PCM.Xだ。CDROM.XはCD-ROMの中身を閲覧し，参照できるもの。もちろん，CD-ROMからコピーもできる。フォーマットは，国際的標準といわれているISO9660フォーマットと，Macintoshのものに対応している。

そして，CD2PCM.X。これは私はよくお世話になる。オーディオCDの再生のほか，なんとオーディオトラックの取り込みができるという優れもの。CD-ROMドライブが東芝製とソニー製の一部に限定されるものの，オーディオCDから綺麗にサンプリングできるのが嬉しい。アナログ信号を通さないから，かなりクオリティの高い録音ができるのだ。ただ，CDは構造上，録音面（すなわち表面に空いたホールそのもの）を，完全確実に読み取ることはできない。それをうまく補正する技術が，オーディオCDメーカーの腕の見せどころだ。何も考えずに読み出すため，精度はCD-ROMドライブに依存する。少し試してみたところ，出回っているソニーのCDU-561系よりも，今回使用したCDS-Eに搭載のCDU-55S系のほうが音がよさそうだ（ただし，プログラムを多少改変する必要がある）。これなら，アナ

ログケーブルを通してX680x0のIN端子より入力されたサンプリングデータよりも格段に綺麗だ。

次は，各種グラフィックローダ。MS-WindowsのBMPフォーマット，国際規格のJPEGローダ（の速い奴），TIFフォーマットのローダなどなど。MS-WindowsのCD-ROMが*ISO9660*ってことは，BMPローダを使えば，MS-Windowsの壁紙集（CD-ROM）なども見られるってことになる。さらにMacintoshのPICTファイルのローダが入っているので，MacFileTransfer.xを使えば，Macintoshのグラフィックも読める。また，これ以外にも，高速な柳沢PICローダであるHAPICや，噂の柳沢PIC2なども入っている。グラフィックに関しては，いろいろなエフェクタも入っていて，なかなか楽しいことができる。

あと，私が作ったPCMPLAY。何に使うかって？　さっきいったとおり，MEGA-CDのCD-ROMはISO9660だから，ゆみみみっくすのCD-ROMでも持ってきてファイルの拡張子をPS8にリネームして，PCMPLAY *.PS8としたら……これは古いバージョンだが，新バージョンはPCエンジンも……とまあ，話が脱線しそうなので，詳細はまたいずれ。

SX-WINDOWユーザーのためのツール

私の身のまわりにも，SX-WINDOWに環境を移行してきた人がかなりいる。しかしながら，通信をしていない人の環境はまだまだ整っていない。やっぱりSXも，フリーソフトで環境が作られつつある。ただ，SXに限っては詳しい人が周りにいないと環境の整備は難しい。したがって，SXのソフトウェアは転載されることが少なく，通信でも手に入りにくい。

そういうことを考えれば，このフリセレ2には，SXのツールが比較的多いので，結構楽しめるのではないだろうか。具体的にいくつかのソフトを紹介してみよう。

まずは，QuTERM。SXユーザーにはもはや常識的な通信ソフトだ。シャープの「CommunicationSX-68K」の不具合がなかなか直されないといった状況もあり，多くのSXユーザー兼パソ通ユーザーはQuTERMを利用している。当然私もだ。現在のQuTERMは，すでにマクロ言語としてLispを搭載し，ますますアプリケーション化してきたが，これに収録されているのはその前の安定バージョンといったところ。メモリもそれほど食わないので，4Mバイト程度の小メモリ環境ではなかなか強いソフトだろう。慣れるまでは最初の設定が難しいかもしれないが，通信ソフトとエディタなどが同時に動く環境は，一度手に入れたら手放せないぞ。

次に，独断と偏見で紹介するのはGRROOT.Xである。Oh!X誌上でもちらほら出てきたから，知っている人もいるだろう。そう，6万色背景のツールである。6万色背景にはほかに，先月号の付録ディスクに収録された石上さんのSXPICS.Xがあるが，あれは，GRROOT.Xと違い，マルチタスクで処理を行うものの，SX-WINDOWに用意された規定の方法でロードされていないため，対応形式はあくまでもPICだけだ。GRROOT.Xが厳密にSX-WINDOWの規定に則って作られているわけではないが，セマフォ管理[*2]などで安全化が図られているし，なにより，IVM.LBに入っている形式のグラフィックなら何でも背景にできるため，私はこちらを使っている。

そのほかにもフリセレ2には，全部は紹介しきれないほどいろいろ収録されている。たとえば，エラーダイアログの変更，ファイル名の補間，ビープ音の変更，マルチピリオドにSXをある程度対応させるもの，X-WINDOWライクにSXのアクティベートをさせるもの，などなど，スタートアップメンテなどに登録して環境を整えるツールがたくさんある。

うーん，みればみるほど，SXユーザーには買いってところかな？

＊2――セマフォ管理とは，いわゆる排他処理のようなもの。たとえば，GRROOT.Xが実画面を使用，つまり背景を表示しているならば，GRW.Xを起動できないといったもの。マルチタスクで動くお互いのプログラムが，限られたひとつの資源（ハードウェアリソース）を使おうとブッキングしないように，本来はSX-WINDOWが持っているべき機能だ。

フリーソフト使用上のお約束

このCD-ROMは，あくまでもフリーソフトウェアの集大成である。収録されているソフトウェアそのものは，大半が無料で無保証だ。だから，うちのドライブではCD2PCMが動かないとか，MacintoshのCDがPICTローダで見られないとか，そういう苦情は，すべて自分自身で処理するのが常識。フリセレ2をまとめた計測技研にも責任はないはずだ。

それから，ときどき勘違いしている人がいるが，フリーソフトはあくまでも無保証だ。だから，次は○○に対応してほしいとか，そういう要望を「押しつける」のは本来はタブー。お願いならばよいが，相手は無報酬で作っていることを忘れずにね。

あと，ときどき雑誌などで「通信をすれば便利なソフトがたくさん手に入ります！」とフリーソフトを餌に通信を勧める記事がある。しかし，これには賛成できない。通信のネットワークの楽しみ方は，や

CD-ROMドライブ

先月号の「C MAGAZINE」など，最近の雑誌にはよくCD-ROMがついている。CD-ROM特集も多く，まさにCD-ROMはいまが旬といった感じ。

事実，CD-ROM並みに「安く」「広く」「量産が簡単なもの」はそんなにたくさんはない。しかし，ときどき，何を勘違いしているのかMOと比較している人がいるけど，あれとこれとは存在意義が違う。MOはデータ記憶媒体。そしてCD-ROMはデータ供給媒体だ。

今回利用したドライブは，メルコのCDS-E。2倍速強程度で低価格。作りはちょっと貧弱ながら，なかなか働いてくれる。

ドライブ自体は，フロントローディング式，キャディは入らないタイプでCDはただ置くだけ。音楽CDの演奏もでき，PhotoCDつまりマルチセッションに対応している。X68000のSX-Photographyもばっちり動く。ただ，なぜかCDPlayer.XとMacFileTransfer.Xが動かない。CDPlayer.XはCD2PCM.Xで再生できるのだから，やっぱりCDPlayer.Xがおかしいのかもしれない。でも，MacFileTransferが使えないのはなぜだろうか。CDROM.XでもMacintoshのCD-ROMがなぜか読めなかったし……。

一説によるとフロントローディングタイプは音がいいそうだが，私には普通のCDと同じように感じられた。ドライブはソニー製で，今回紹介したフリセレ2のCD2PCMも，取り込みはベンダチェックを少しゆるくするだけで動く。総合的にみてドライブとX68000との相性が悪くないという程度。とにかく安いからねぇ。

CD-ROMが必要か否かは，MOとの比較などという問題ではなく，データ供給メディアに投資するか否かの問題だと私は思っている。X68000用と謳われたソフトでなくとも，雑誌の付録などにも流用できるデータがあるし，グラフィックの，いわゆるセクCDや，いやらCDは存分に利用できる（ただし，MAC用は工夫が必要）。

いずれにしても，特に通信をしていないユーザーにとっては有意義なメディアだろう。

私は，X68000ユーザーにCD-ROMはイチ押しだ！と思っている。個人的な最初の目標は，U氏を口説き落として，CD-ROMの特集だな。その次はOh!Xの付録CD-ROM。付録ディスクの集大成以外に，入らなかったソフトをたんまり入れられる。最終目的は，X68000ユーザーにCD-ROMは必須という状態。ただし，これはあくまでも私の個人的な目的なので，編集部の意向とは無関係です。勘違いしないようにね。（瀧）

CDS-E　29,800円（税別）
メルコ　☎052(619)1827

っぱり，仲間を増やす楽しみが「1」で正しいはず。もちろん，フリーソフトだけが目当てでも，誰にも迷惑をかけなければ悪くはない。ただ，私は何も作れないからといって逃げないで，不具合の報告などをするのも立派な通信での交流だと思う。

今回のフリセレ2は，作者に直接感想などはいえないが，もし，あなたが通信をしていたり，これからしようと思っているならば，そういうことは忘れないでほしい。

感想

なかなか楽しめるCD-ROMでした。普段，muleなどは簡単にはダウンロードできませんからねえ。

そうだな，あとは満開製作所のベストor集大成版が欲しいなあ。私は最初のほうの電脳倶楽部は持ってないし。やっぱり，満開製作所にお願いのお手紙作戦か？

本当は，Oh!Xの付録CD-ROMをなんとか作ってほしいところ。毎回入れるものはたくさんあるのに，全然入らないんだもん。SLASHとSX-BASICの完成版をCD-ROMで配布！　っていうのもいいと思うんだけどなぁ。

CD-ROM Driver ver.2.00

「Free Software Selection Vol.2」の発売に合わせてなのか，「CD-ROM Driver」がバージョンアップした。これにより，SCSI-IIに対応し，Human68k ver.3.0に完全対応し，付属ツールの一新が図られている。

まず，最初のSCSI-II対応の意義だが，これはかなり大きい。SCSIでは，CD-ROMは，「リードオンリーデバイス」として扱われてはいたものの，限定はされておらず，音声やビデオのいわゆるオーディオ情報の伝送をはじめとする細かい機能は規定されていなかった。したがって，CD-ROM固有のオーディオデータの演奏などは，メーカーごとに異なっていたのである。

SCSI-IIになって，リードオンリーデバイスは「CD-ROMデバイス」となり，データは別ケーブル（AUXケーブル）で伝送するものの，オーディオ情報の操作などのCD-ROM特有の操作もSCSI-II規定方式で命令できるようになった。

簡単にいえば，CD-ROM Driver ver1.00の時期は，計測技研のドライブ専用で，ver.2.00になって，汎用になったという感じである（厳密にはちょっと違うけれど）。

だから，SCSI-IIの規格のものならば，どんなドライブでもたいていは動くはずだ。前バージョンでもISO9660フォーマットのメディアはSCSIで扱えたが，音楽CDやCD-ROM XAメディアへのアクセスはSCSI-IIのみの対応だ。

そういうわけで，CD-ROMを購入するなら，CD-ROM Driverもver.2.00を買うべきだ。店頭にはまだver.1.06もあるようなので注意してね。

それから付属ツール。これには，まず，SXのデスクトップで音楽CDを演奏するCDPlayer.x，そしてこれらを自作プログラムで起こすためのライブラリ，それから，MacintoshのCD-ROMから読む，MacFileTransfer.xがある。なぜか，この2つはCDS-Eでは動かなかった。どっちが悪いのかは謎。まあ，普通に考えればソフトのほうに原因がありそうだが。いずれにしても，早いうちにサポートしてほしいところ。せっかくの安いドライブが使えないのは残念だ。

ここで余談。

SCSI-IIはX68000に接続できますか？　という質問がときどきあるけれど，これは大丈夫。そもそもSCSIには，MOも規格に定められていない。SCSI-IIという規格は，SCSIと互換性を持たせつつしっかり規格化を図り，さらに拡張したもの。拡張機能を使っていない限り，SCSIでもSCSI-II機器は問題なくつながるはず。これはハードディスクにしても同様。怪しげなSCSIよりも，SCSI-II機器のほうが安心できる。

もっとも，それでも，うまくつながらないことがある。X68000のせいか，ターゲットが悪いのか，それとも単に相性が悪いだけなのか。このあたりは，SCSI関連の書籍を読んでも全然わからないことがあり，失敗もままある。SCSIはなかなか奥が深いのだ。　（瀧）

CD-ROM Driver ver.2.00　　4,800円（税別）
計測技研　　☎0286(22)9811

収録ソフト一覧

```
¥
├─ BSD_IMG      NETBSDのMOイメージ
├─ BSD_INST     NETBSD関連
├─ CGD          CGデータ集
├─ CGT                以下はCGツール
│   ├─ AZOOM     拡縮プログラム
│   ├─ BMPL      BMPローダ
│   ├─ CO        データタイプ変換
│   ├─ DCOLOR    減色プログラム
│   ├─ EX_COPY   EX-WINDOW用画面合成コピー
│   ├─ G         グラフィック閲覧ツール
│   ├─ GRCNV     640x400(16)→512x512(65535)
│   ├─ GRON      CG画面のON
│   ├─ GVIEW     CG拡大VIEWER
│   ├─ HAPIC     高速柳沢PICローダ
│   ├─ HMAGL020  高速MAGローダ
│   ├─ JPEGED    高速JPEGローダ
│   ├─ MFGED     グラフィックツール
│   ├─ MMVP      アニメーションツール
│   ├─ MOVEIT    同上
│   ├─ PCT_PIC   PICTローダ
│   ├─ PIC2_008  柳沢PIC2ローダ
│   ├─ PIL103    超piローダ
│   ├─ SZCONV    ｽｰﾊﾟｰﾀﾞﾌﾞﾛｰ(.RGB)→.ZIM
│   ├─ TIF022    TIFローダ
│   ├─ VTOOL     イメユニ取り込みツール
│   └─ ZSCONV    .ZIM→ｽｰﾊﾟｰﾀﾞﾌﾞﾛｰ(.RGB)
├─ ETC                その他
│   ├─ ACTA      芥川龍之介PDD集
│   ├─ MUNIT     ﾏｰｷｭﾘｰﾕﾆｯﾄの回路図
│   ├─ PIC_FMT2  PICのフォーマット解説
│   └─ PITECH    PIのフォーマット解説
├─ FONTS        フォントいろいろ
├─ HU                 以下はHuman上のツール
│   ├─ ACK132X2  日本語コードコンバータ
│   ├─ AISH108X  ISHコンバータ
│   ├─ ALARM     アラーム
│   ├─ ASTGARD   RPG Like BBS System
│   ├─ BADFMT    不良セクタ発見ツール
│   ├─ BDIFSET   バイナリ差分ツール
│   ├─ BLVPX01A  レスポンスもできる、賢い?ページャ
│   ├─ CASL      いわゆるCASL
│   ├─ CCOPY24B  収集家の為のコピーコマンド
│   ├─ CD2PCM    CD音声取り出しツール
│   ├─ CDROM     CDROM閲覧ツール
│   ├─ CHAKOSHI  テキストフィルタ
│   ├─ CHGSYS10  環境選択プログラム
│   ├─ CONTRAST  画面焼付防止ドライバ
│   ├─ CVS       プログラム開発支援ツール
│   ├─ DASH108   UNIX シェルを真似たシェル
│   ├─ DCA       オルタネートディスクコピー
│   ├─ DCACH212  ディスクキャッシュ
│   ├─ DCTRL100  drvctrl
│   ├─ DDIR      ディレクトリ容量表示コマンド
│   ├─ DEDIT     ディスクエディタ
│   ├─ DJPRINT   カラー印刷ソフト
│   ├─ DMS       Double Mouse System driver
│   ├─ EDR099XI  エディタ(EDコンパチ)
│   ├─ EDT       エディタ(EDコンパチ)
│   ├─ EJ        和英，英和辞典(辞書無)
│   ├─ EM_R5B    μEMACS
│   ├─ EXTDRV16  強制終了プログラム
│   ├─ FEE111    電話料金算出
│   ├─ FLT4LIB   (X68030&68882専用ライブラリ
│   ├─ FMT       ディスクフォーマット
│   ├─ FONTED2   フォントエディタ
│   ├─ FSCK103   ディスク修復ツール
│   ├─ GRU       目の錯覚を楽しむツール
│   ├─ GSSRMD10  外字,ｽﾌﾟﾗｲﾄ,SRAMを纏めてRAMDISKに
│   ├─ GZIP      GNU ZIP(COMPRESSER)
│   ├─ HAS503    ハイスピードアセンブラ
│   ├─ HFONT120  12ドットフォント
│   ├─ HLK229B   ハイスピードリンカ
│   ├─ HSIPL216  高速IPL
│   ├─ IBMDRV07  IBM FORMATドライバ(MO)
│   ├─ ISSCD     プログラム開発ツール
│   ├─ KEYLOCK   キー入力禁止／解除
│   ├─ KOBITO10  小人さん
│   ├─ KSHX25F   K-SHELL
│   ├─ LHA_X640  LHA(アーカイバ)
│   ├─ LL        プリンタユーティリティツール
│   ├─ LOGCUT    ログカッター
│   ├─ MAKEX12F  MAKE(開発支援)
│   ├─ MINT      ファイルメンテツール
│   ├─ MIXSIX11  キーワード検索
│   ├─ MOCOPY10  MOコピーツール
│   ├─ MULE      mule
│   ├─ NCV       プログラム開発支援ツール
│   ├─ NSCAN     Scanner Driver for CZ-8NS1
│   ├─ OFFTIMER  オフタイマー
│   ├─ OLEO      Oleo 1.5
│   ├─ PAPERDRV  起動ドライブをAドライブと交換
│   ├─ PLAIN2    TXT → TeX
│   ├─ POST      郵便番号検索システム
│   ├─ PULLUP02  テキストファイル救出ツール
│   ├─ RANDOM5   ランダム抽出
│   ├─ RCS       RCS(開発支援)
│   ├─ RDEX1     ランダム実行
│   ├─ RKDRV     キー入力改善ドライバ
│   ├─ RSBUF100  RSDRV.SYS バッファ設定ドライバ
│   ├─ SCRC      inquiry と read capacity
│   ├─ SGRAMD    SYSTEM外字-RAMDISKドライバ
│   ├─ SHAKE206  ディレクトリの空き詰め
│   ├─ SIMSP86   大宇宙の神秘シミュレータ
│   ├─ SPRMD     スプライトRAMDISKドライバ
│   ├─ SSRAMD    SRAMDISKのいいやつ
│   ├─ STEVIE    vi「を目指した」エディター
│   ├─ T2Z113    ザウルス用ﾃｷｽﾄﾌｧｲﾙｺﾝﾊﾞｰﾀ
│   ├─ TARGET    X68000をターゲットに
│   ├─ TAR_M0    TAR(アーカイバ)
│   ├─ TK_IPL    Original I.P.L. Program
│   ├─ TMNU      高性能な通信ソフト
│   ├─ TMSTORE1  ﾀｲﾑｽﾀﾝﾌﾟ保護ﾌﾟﾛｸﾞﾗﾑ
│   ├─ TM_CTL    RS232C 通信パラメータ表示と設定
│   ├─ TWON      TwentyOne.X
│   ├─ VIMX68K2  Vim version 2.0
│   ├─ ZCOM101   ザウルスとのデータ通信プログラム
│   └─ ZPRN100   ザウルス用プリンタ出力ツール
├─ KO           KO-WINDOWのツール集
├─ MUD          音楽データ集
├─ MUT                以下はMUSICツール
│   ├─ APREC101  ADPCMの録音・再生が出来ます
│   ├─ MMDSP029  ヴィジュアルプレイヤー
│   ├─ MMX102    RC-system 用ﾏﾙﾁMIDIｾﾚｸﾀ
│   ├─ MSOFSNC0  PowerOffSynchronizer
│   ├─ MUSTOOLS
│   ├─ OPM2ZMS   OPM → ZMS
│   ├─ PCM8A     PCM8A
│   ├─ PCMPLAY   PCMプレイヤー
│   ├─ PCMREPL   ADPCM長時間録音・再生プログラム
│   ├─ RCSYSTEM  RCのシステム
│   ├─ STED203W  RCのステップエディタ
│   └─ ZMUSIC    ZMUSICのセット
└─ SX                 以下はSXのツール
    ├─ AXACTJP0  ｱｸﾃｨﾌﾞｳｲﾝﾄﾞｳを中央に
    ├─ BEEPCHG   BEEP音を変えるツール
    ├─ CALC      式入力計算器
    ├─ CALEN104  カレンダー
    ├─ CARD3     トランプカードデータ
    ├─ CCCV100   CCCV(開発支援)
    ├─ EDHOK101  エラーダイアログ変更
    ├─ FCMPL     ファイル名補間
    ├─ FIXSEMB   SX Ver3.0のセマフォをFIX
    ├─ G2TSERV   G→Tの変換サーバ
    ├─ GABAG040  ヒープ監視&管理ツール
    ├─ GNUPLT    GNU PLOT
    ├─ GRRT120   6万色背景
    ├─ GSM020    gScript作成ツール
    ├─ IWDF010B  アイコン用WDEF
    ├─ JIS064    コード変換プログラム
    ├─ JIS2_121  第2水準漢字のセレクタ
    ├─ MINSX110  マインスイーパー(ゲーム)
    ├─ MKHIME10  日めくりカレンダー
    ├─ MOVIE082  アニメツール
    ├─ MSX521    SXの開発ライブラリ
    ├─ PAIMN101  麻雀牌のリソース縮小版
    ├─ PAIST101  麻雀牌のリソース
    ├─ PERI0042  SXをマルチピリオドに
    ├─ PICBTTN   ピクチャーボタン
    ├─ PITMAN    ゲーム
    ├─ QU        通信ソフト
    ├─ RSCV121   リソースビュア
    ├─ SCLOCK5   小時計
    ├─ SM        ストリームマネージャ
    ├─ SMAEX01   タスクの起動と接続
    ├─ SMMINOA1  スクリーンセーバーモジュール
    ├─ SMTEXT02  ストリームテキスト
    ├─ SNAP01    Snap-to
    ├─ SXACT     X-WINDOWライクなアクティベート
    ├─ SXAD003   Auto Dir
    ├─ SXBACK02  背景設定
    ├─ SXBIO     バイオリズム
    ├─ SXBUP100  BUP
    ├─ SXCHN140  青牌(パズルゲーム)
    ├─ SXCUT01   画像カッター
    ├─ SXG2T21   G→Tの変換
    ├─ SXGAL     画像閲覧
    ├─ SXGCH     GNU CHESS
    ├─ SXGZIP06  GZIP
    ├─ SXLN000   LN
    ├─ SXMEMO10  メモ帳
    ├─ SXMKTC00  SXMOVIE用のツール
    ├─ MLINK3    MLINK(通信プロトコル)
    ├─ SXMODE04  画面モードチェッカー
    ├─ SXNEKO    ねこ
    ├─ SXPIXP13  PIXとPIの閲覧ツール
    ├─ SXTAR030  TAR
    ├─ SXTIMER   タイマー型アクセサリ
    ├─ TASK109   TASKモニタ
    ├─ UNO       UN○のカードデータ
    ├─ VISON104  LZH/TARのビュアー
    └─ ZXC112    ウインドウロケータ
```

Graphic Gallery

今月のGraphic Galleryは，本文とは直接関係ありませんが，8月号のGraphic Galleryに掲載した森山さんの恐竜を特集します。

この恐竜は，「SWORD」「SWORD 2」の作者・森山知己氏がCGAシステムver.2で作成したものです。誌面では残念ながらアニメーションの様子は紹介できませんが，非常になめらかに恐竜が動きます。その，尻尾や首などが関節の継ぎ目がなくグネグネと動くのは，新ツールYAWARA.Xの機能です。実は，このYAWARA.Xは，森山さん自身が考案されたツールです(プログラミングはMOOG寺田)。皆さんも，こんなツールが欲しいなどという要望(アルゴリズム明記)を出せば，当方で開発を検討しますよ。

森山氏は，「YAWARA.Xを使えば，誰にでもこんなCGができるんだ。X68000＆CGAシステムはすごい！」とおっしゃっていますが，当方は「こんなんできるんは，あんただけやがな」とつっこんでおきます。

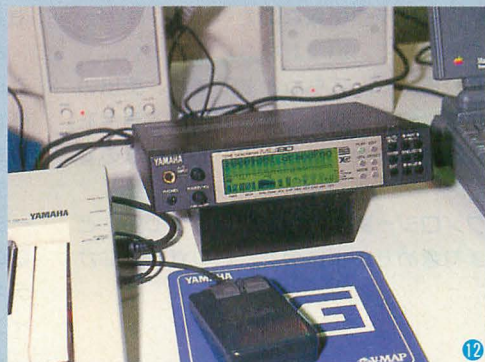
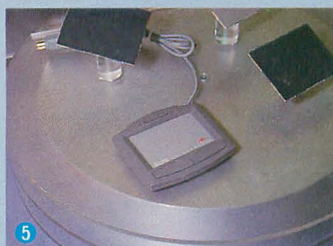
昨年公開された映画「ジュラシックパーク」を見た方は，「こんなCGがパソコンで，個人でできるようになるのはいつのことやら……」と思ったでしょうが，思ったより早そうですね。

この恐竜は，約5673ポリゴンです。さすがに面数が多いですね。YAWARAなどを使っていることもあって，作画にはさすがに時間がかかります。1フレームにXVIで約18分，040turboで2～3分かかってしまいます。

背景の画像は，「MacUser」編集部のAさんにAMIGAで制作してもらいました。

CGAは，マシンパワーではない，アイデアと努力だ。

データショウ'94

DATA SHOW '94

❶アイワのマルチメディアパソコン ❷視力障害者のための読書機で新聞を拡大表示 ❸トークンリング上でサーバからの動画データを表示するOS/2のデモ ❹リモコンマウス感覚のスペースポイント ❺撫でて動かすグライドポイント ❻注目の光ディスクPD ❼ようやく姿を見せたMDデータドライブ ❽なぜかFDインタフェイス対応の低価格ストリーマ ❾MOアクセラレータ ❿オリンパスの高速230MバイトMO ⓫パーソナルになったディスクアレイシステム ⓬ヤマハのXG規格対応音源MU-80

9月26日〜28日の3日間，東京晴海の国際見本市会場でデータショウ'94が開催された。不況は峠を越したと政府はいうものの，今回のデータショウも昨年と同程度の少々寂しい規模での開催となった。

世相を反映してか各社ともマルチメディアが前面に押し出されている。ネットワークを利用した業務システムや画像通信などは毎年発表されているのだが，昨今のInternetや情報ハイウェイなどの話題のおかげでようやく耳目を集めるテーマとして認められてきたといえる。

会場にはハンデキャッパーのための機器を集めたコーナーなどが設けられており，拡大投影型の読書器や点字ディスプレイ，マウスエミュレータなどが展示され，人に優しいパソコンのあり方を見せている。

コンピュータ関係で目立っていたのはアイワのマルチメディアマシン。コンポよりのデザインというのだろうか？ 驚異的なのはIBMで参考出品のラップトップマシン。PowerPCを積み，TFT液晶で1600×1024ドット，12.5型の大画面だ。

別の意味で驚いたのはアイワのPC-9801やPC/AT用の39,800円という低価格ストリーマ。残念ながらFDインタフェイス対応（なんじゃそりゃ）でまたびっくり。

もうひとつの謎。ようやく現れたソニーのMD DATAはなぜかポータブル機専用。さらに内蔵型のものもアメリカでのみ発売予定という。

そのほか記憶装置関係では，普及価格帯に近づいたディスクアレイシステムなどが今後のトレンドになるのだろうか。

注目は松下のPD。これは相変化型光ディスクと4倍速CD-ROMを一体化したもの。ひとつのドライブで両方のメディアをアクセスできる。光ディスク時は120mm径のゾーンCAV式でCD-ROMと同等の650Mバイトを記録する。回転数約2000rpmで平均データ転送速度は870Kバイト/秒とマルチメディアの主流になるか？ と思わせる仕様。原理的にはレーザーの熱の差で記録膜を結晶化させたりアモルファス状にすることで反射率を変えて情報を記録するものだ。データ消去は書き込みと同時に行われるなどの特徴もあわせ持つ。製品化は来春でドライブは10万円前後となる模様。でもCD-ROMと光ディスクって同時に使いたいこともあるよなあ，きっと。

ちょっと変わったところでは，音楽関係でヤマハの新音源規格XGに対応したMU-80が展示されていた。XGはGM規格に拡張を施したものでローランドのGSと似たような位置付けと考えていいだろう。AWM2音源で32チャンネルに対応している。音色エディットなどでどの程度の互換性が保証されるのか興味深いところだ。

［第42回］

うなぎのぼりのハイテク係数

江口　響子

響子inCGわ～るど

涼しくなりましたね。この秋は，おいしいものをたくさん食べよう！　と決心を固めています。今年の夏はあまりにも暑く，食欲がまったく出なかったので，その反動なのかもしれません。皆さんは，いかがでしょうか。

エンゲル係数とは生活費のうち食費の占める割合である，というのはよく知られています。食欲の秋ならば，さぞエンゲル係数が高くなっているのでは，と思いきや……ここのところ，うなぎのぼりなのは，ハイテク係数なのです。

ハイテク係数って，いったいなに？

そう思われるのも，ごもっともですね。支出のうちハイテクノロジー関連のものが占める割合を，まあ，これは自分で勝手に呼んでいるだけなのです。

ハイテクノロジー関連とひと口にいっても，どのあたりまで含めたらよいのか，これがなかなか難しい。いまや身の回りのもので，集積回路をなんらかの形で搭載しているものが，少なくないからです。この夏とてもお世話になったクーラーをはじめ，冷蔵庫，電子レンジ，炊飯ジャー，湯沸かしポット，洗濯機，CDラジカセ，テレビ，ビデオ，LDプレイヤー，パソコン，プリンタ，スキャナ，CD-ROMドライブなどなど。

で，それら全部をハイテクと呼ぶと，収拾がつかなくなるので，コンピュータとその周辺機器，およびサプライ用品までを範疇としました。

さて，それでは，なぜハイテク係数が増えたのでしょうか。振り返ってみると，どうも仕事の中身が，最近変わってきたせいだということに気がつきました。

もともと仕事で，文章を書いたり，イラストレーションを描いてきました。その延長線で，静止画のCGを制作してきたのですが，最近はそれにテキストや動き，音が加わったものを考え，デザインする仕事が増えてきました。

というのも，パソコンの性能が格段にアップし，これらのデータをまったく同質のものとして，手軽に扱えるようになったからです。紙を切ったり，貼ったりするような感覚で，文章やサウンド，CGをコラージュできるのです。そんなわけで，去年の暮れに買ったMacintoshのColorClassic IIが，いまは，Quadra840AVになってしまいました。

また，仕事の内容が少しずつ違うので，そのつど必要なソフトや周辺機器を買い足さなくてはならない場合もあります。MOドライブを買ったのは，マルチメディアソフトのプロトタイプをはじめてデザインしたときのこと。作っているうちに，データ量がどんどんふくらんでしまい，しかたなく秋葉原に買いに走ったのでした。もっとも，グラフィックもサウンドもデータ量は大きいので，当然前もってわかりそうなものですけれどね。そ

のときは，なんとかなるわい，と思って作りはじめてしまい，あとであわてたのでした。なさけない話です。

そんなこんなで，私のハイテク係数はとんでもなく高くなってしまい，財布と通帳を見ると，なんだかためいきが，ふぅ～と出てしまいます。おいしいものをたくさん食べよう！　という決心は固いのですが，このままだと，B級，C級グルメで終わりそう……でも，まあいいか。

実は最近，仕事場から1～2分のところに，おいしいそば屋さんを見つけたのです。こんな近くにあったのになんとなく入らずにいたのが，不思議なのですが。

その店では，カツ丼がおすすめ。分厚いお肉にからりとした衣，甘すぎないタレと卵が，ほかほかのごはんとうまくからみ合って……ああ，おなかが空いてきた。

今回のCGデータ

1280×1024ピクセル
1670万色フルカラーを4×5ポジで出力
X68000用のサイクロンで作成したRGB画像を，XIN/XOUTIII（電机本舗）でMacintoshに転送し，Photoshop2.5Jでコラージュ
「響子in CGわ～るど」のCGは，ピクセルの縦横比（アスペクト比）1：1でレンダリングしたものを，ポジ出力しています

SOFTWARE INFORMATION

ビデオゲームアンソロジーシリーズの11弾はナムコの「パックランド」に決定しました。シャープの「書家万流SX-68K」にも期待が寄せられていますが，そちらはまだ情報なし。ごめんなさい。来月はどうかな？

魔法大作戦

編集部へのはがきのなかでも期待度の高い縦スクロールシューティング「魔法大作戦」。

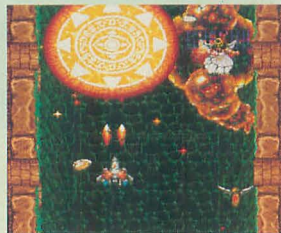
4人の個性的なキャラクターからひとりを選び，平和な王国に来襲した亜人国家の帝王ゴブリガンを倒すというストーリーだ。

キャラクターを少し紹介しよう。まずは，猿を肩に乗せた強腕戦士ガイン。自称，無敵の戦士。次は，魔導戦闘機でド派手な攻撃をする魔法使いチッタ。彼女はかつて世界を救った大魔導師を祖父に持ちながら，田舎の生活を嫌って都にとび出したおてんば娘である。3人目は，巨人忍者(ジャイアントニンジャ)に殺された師匠の仇を討つために旅する侍竜(サムライドラゴン)ミヤモト。竜人なので生身での戦いだが，攻撃力はもちろんほかの3人にも劣らない。そして最後は，骨収集家の呪術師ボーンナム。闇の魔導師である彼には悪い噂がつきまとうという。この4人，攻撃力もスピードも異なるので，誰を選ぶかはお好み次第。

ゲームの詳細はまだわからないが，編集部に順次届けられる開発途中バージョンを見る限りでは，なかなか期待できそうだ。

発売予定は12月。

X68000用　5″2HD版　9,800円(税別)
EAV　☎03(5410)3100

画面は開発中のものです

上海 万里の長城

麻雀牌を使ったパズルゲーム「上海」。その新作「万里の長城」は11月に発売される。

対戦型の「バトル上海」のほかに，ひとりプレイにはルールの異なる3つのモードがある。それぞれ，プレイヤーは万里の長城の始点・山海関から終点・嘉路関までをパズルを解きながら進んでいく。どのモードでもステージ数は40面あり，3ステージごとにルーレットやサイコロ振りなどのボーナスゲームもあるそうだ。

図柄も麻雀牌以外に星座や音符などが選べ，カーソルの形も変えられるので，好みに応じて違った雰囲気でプレイできるだろう。

X68000用　5″2HD版　8,800円(税別)
EAV　☎03(5410)3100

パックランド

電波新聞社のアンソロジーシリーズ最新作は，1984年にナムコからリリースされた隠れた名作「パックランド」だ。

パックマンの国・パックランドに迷いこんだフェアリーを，プレイヤーの操るパックマンが数々の障害を乗り越えて，妖精の国まで送り届けるというストーリー。いわゆる横スクロールアクションゲームだが，すべての動作を3つのボタンだけで行う非常に斬新な操作システムで，また，アメリカで好評であった「TVアニメ版パックマン」を原作としていて，当時としては珍しいフルカラーの美しい背景のなかをアニメチックなキャラクターが動き回る人気ゲームだったのである。また，ジャンプタイミングで特典が決まる「芸術点」やさまざまな隠しフィーチャー，それに完成度の高いBGMなどにより，一部マニアの間でも評価は高かった。

楽しくも深くも遊べる古典ゲーム，パックランド。12月の発売予定だ。（で）

X68000用　5″2HD版　7,800円(税別)
電波新聞社　☎03(3445)6111

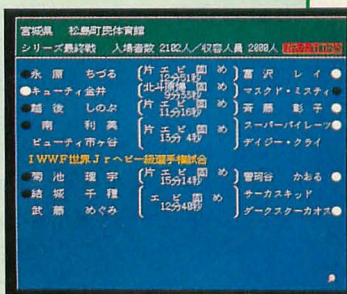
レッスルエンジェルスSPECIAL

人気シリーズの最終作「レッスルエンジェルスSPECIAL」。開発途中バージョンが入手できたので，今月掲載したのはすべてX68000版の画面である。

かつて日本マット界に君臨していたブレード上原を師匠とする主人公は，小さな独立団体「太平洋女子プロレス」を支え，世界の王者をめざして戦い続けていく。相手は前作などでおなじみのあの日本人レスラー，そして世界の強豪たちである。

ワールドサーキットモードで起こるさまざまな出来事に対しては，いくつかの選択肢があり，そこでどう行動するかで主人公の運命は変わっていく。

水着剝ぎデスマッチもあり，今度は18禁，パッケージ販売のみ。12月発売予定。

X68000用　5″2HD版　8,800円(税別)
TAKERU　☎052(824)2493

VIEW POINT

シューティングは激ムズでなければつまんない，なんて贅沢者の方々はもうしばらくお待ちください。クオータービューのシューティング「VIEW POINT」の発売は11月の予定です。

敵キャラクターはバイオメカニカルなイメージです。昆虫嫌いとか魚はダメ，とか甲殻類が憎いなどの性癖をもつ人には，闘争心がかきたてられてよろしいかもしれません。

難易度はコンフィグ画面で変更できるので，ぬるいアナタにも楽ししめるので，ご安心を。

X68000用　5″2HD版　7,800円(税別)
ネクサス インターラクト　☎03(5474)3581

画面は開発中のものです

発売中のソフト

★XDTP SX-68K　シャープ
X68000用　3.5/5″2HD版　35,000円(税別)
★スーパーストリートファイターII　カプコン　9/30
X68000用　5″2HD版　9,800円(税別)

新作情報

★クイーン・オブ・デュエリスト外伝α＋LIGHT　TAKERU　11/1
X68000用　3.5/5″2HD版　5,800円(税込)
★上海 万里の長城　EAV　11/26
X68000用　5″2HD版　8,800円(税別)
★VIEW POINT　ネクサスインターラクト　11/未
X68000用　5″2HD版　7,800円(税別)
★パックランド　電波新聞社　12/上
X68000用　5″2HD版　7,800円(税別)
★魔法大作戦　EAV　12/16
X68000用　5″2HD版　9,800円(税別)
★レッスルエンジェルスSPECIAL　TAKERU　12/中
X68000用　5″2HD版　8,800円(税別)
★X CASE　Béシステム
X68000用　5″2HD版　19,800円(税別)
★ロボスポーツ　イマジニア
X68000用　5″2HD版　価格未定
★Traüm　象スタジオ
X68000用　5″2HD版　価格未定
★鮫！　鮫！　鮫！　KANEKO
X68000用　5″2HD版　価格未定
★達人　KANEKO
X68000用　5″2HD版　価格未定
★エアバスター　KANEKO
X68000用　5″2HD版　価格未定
★サバッシュII　ポプコムソフト/クローディア
X68000用　5″2HD版　価格未定
★麻雀悟空・天竺への道　シャノアール
X68000用　5″2HD版　9,800円(税別)
★スタークルーザーII　アルシスソフトウェア
X68000用　5″2HD版　価格未定
★地球防衛MIRACLE FORCE　カスタム
X68000用　5″2HD版　価格未定
★プリンセスメーカー　ニュー
X68000用　5″2HD版　14,800円(税別)
★ディグダグ/ディグダグ2　電波新聞社
X68000用　5″2HD版　価格未定
★フォント＆ロゴ　デザインツール　書家万流SX-68K　シャープ
X68000用　3.5/5″2HD版　価格未定

闘いへの扉はいま，開かれた

Nishikawa Zenji
西川 善司

先月号ではサンプル版にてご紹介した「スーパーストリートファイターII」がとうとう店頭に並びました。対CPU戦，友達との対戦，トーナメントと，いろいろ楽しんでいるでしょうか。今月は製品版をみてみます。

ついに発売されたX68000版「スーパーストリートファイターII」。今月は，移植完成度に焦点を絞り「オリジナル」VS「X68000版」という視点で紹介しよう。

「スーパーストリートファイターII」(以下スパII)は，アーケードでは意外に短命のゲームだった。出たときはそれなりに注目されたが，すぐ，SNKの「餓狼伝説SPECIAL」に食われ，それに対抗するかのように「スーパーストリートファイターIIX」が登場。私の周りの，ゲームセンターにたまには行く程度のそんなにゲーマーゲーマーしていない友人にはスパIIの存在を知らない者もいる。そう，ストIIターボの次は「X」だと思っている連中もいるのだ。

ここで念のためにいっておくが，今回発売のX68000版スパIIは「X」ではない。だからディージェイは蹴り昇らないし，SUPERゲージもない，あしからず。

BGM対決

このストIIシリーズは，「ストリートファイターII'TURBO」までは，CPシステムと呼ばれるカプコンのアーケードゲームシステム基板上で動いていた。CPシステムは4オペの-FM音源 8ch＋PCM音源が数chという音源構成で，X68000の音源構成に似ていた。そのため細かい部分での問題はあったものの，X68000版ストII'はかなりオリジナルライクなサウンドが楽しめた。

ベガよ，泳ぐなら海へいけ。

が，スパIIになってCPシステムもCPシステム2へと一気にハードがパワーアップ。音源も全チャンネルPCM(？)音源で，それに伴って，曲は同じストIIでも，音質/音色は格段に「スーパー」となった(アレンジが変だという意見もあるが)。

で，X68000版ではどうなるのか不安だったのだが，ちゃんと「スーパー」用に新しくなってる。が，今回の移植では内蔵音源によるBGMはオリジナルそっくりとはいいがたい。そりゃそうだ，なにしろ音源が違うのだから。ではどうなったかというと，X68000の内蔵音源である，FM音源＋AD PCM音源で「オリジナルのサウンドをなんとか再現してやる！」と頑張っているのがわかるような感じ。もちろん，オリジナルサウンドを収録したCDなどと聴き比べれば明らかに音は別ものなのだが，決して聴き劣りはしない出来栄えだ。X68000の音源はストII'のときから変わっていないし，曲も同じ「ストIIの曲」なのだが，X68000版のストII'とスパIIの曲を聴き比べてみると明らかに後者のほうがカッコいい。

一方，MIDIの音楽は期待はすれど，ストII'でクソボロだったから今回も……と不安がよぎる。前作ストII'では，MIDI/GM対応とありながら，どのGM音源でもヘボい演奏だったからね。しかし，ご安心あれ。今回はマトモ。SC-55系GS音源でもちゃんと鳴っている(SC-55mkIIがベスト？)。アレンジはかなりスパIIアレンジに忠実。単なる「エヘヘ楽譜どおりに打ち込みましたぁ」という手抜きデータではない。曲調に合ったテンポ/リズムの溜めやベロシティの揺らぎなどかなり入念に作られていて，美しいゲーム画面に負けていない。

音楽面は，オリジナルとの音源の相違から，完璧とはいえないが，別の角度から向上をめざしているので「互角」としよう。

サウンド対決

X68000版では，本体CPUが68030だと効果音が多重AD PCMドライバにより同時発音数が4ボイスへと拡張される。68000では単声だ。単声の場合，マイキャラが必殺技名を叫んでも敵の叫び声にかき消されたりしてしまう。BGMをMIDIで鳴らせばまだましだが，内蔵音源だと，BGMのアクセントであるAD PCMリズムパートまでもが効果音を邪魔し，せせこましさが倍増する。

68030機では，起動時に4ボイスAD PCMドライバが自動的に組み込まれ，効果音とパーカッションが最大同時発音数4まで重なる。X68000版ストII'の4ボイスAD PCMドライバは，よく聴くと挙動がおかしかった。たとえば，マイキャラが波動拳を撃った直後に体力0で負けたとしよう。すると，「はどーけん」の叫び声に続き，断末魔「うーわー」が鳴る。そして「はどーけん」と「うーわー」が重なって和音となってしまうのだ。これでは不自然。普通は「は

わー足に火がついた誰か消してよぉ

X68000用　5"2HD版　9,800円(税別)
カプコン

ど ー，う ー わ ー」となるはずだ。AD PCMドライバは，単純な後着優先発音方式を採用していたようだ。

X68000版スパIIではこれを大幅に改善。プレイヤー1キャラ用に1ch，プレイヤー2キャラ用に1ch，音楽用に1ch，効果音用に1ch，といった具合に用途別にチャンネルを振り分けているようで，オリジナルライクな効果音演出がなされている。

効果音の音そのものについては，特に不満はない。ストII'で不評だった意味不明に加工してある音とは違い，オリジナルそっくりのモノが鳴っている。ただオリジナルでは断末魔の声にエコーがかかっていたのだが，それが省略されている。

この効果音勝負はX68000版の「エコー」なしの事実発覚で，オリジナル版の勝ちということにしておくが，それほど重要な問題ではないだろう。

グラフィック対決

驚くべきことに，グラフィックに関しては，まったく違いがわからないほどそっくりに再現されている。解像度もオリジナルとほぼ同じ。お見事。解像度に関しては他メーカーにもぜひ見習っていただきたい。

各キャラクターの動画も，スーパーファミコン版のようにハショられてはいない。さすがディスク7枚組，さすが容量に糸目をつけないパソコンソフトといった感じ。あのリアルなリュウが波動拳を撃つオープニングもバッチリ。フルスペックだ。

もうひとつ驚かされたのは，背景動画がことごとく再現されている点だ。たとえばディージェイのステージの背景動画。あれだけのダンサーたちが踊りまくり，さらに背景が3～4重に重なりあっている。もう何がBGで何がグラフィックで何がテキストなのか皆目検討もつかない。……スゴイ。

最新鋭のアーケードハードで動いているゲームのグラフィックが，そのまま再現できてしまうX68000。「5年先を見越した設計」というのはあながちハッタリではなかったことを再認識させられた。

細かいところを探せば，キャミーステージのオーロラの色が違うとか，「求む対戦プレイ」の文字がきらめかない……といくつか挙げられるが，いわれなきゃ気づかない程度なのでよしとしよう。

この勝負は引き分けだろう。

処理速度対決

ゲームスピードはXVI系16MHzだと完璧。遅くならない。背景動画バリバリのステージでも快適だ。一方10MHzでは，背景動画の盛んなステージや，キャラクター同士の派手なアクションが度重なるとググっと重くなる瞬間がある。ただし，「これは10MHzだから……」といい聞かせれば，十分プレイできる。ひと昔前のX68000用3Dポリゴン系ゲームと比べればへっちゃらへっちゃら，重さは気にならない。ただし，16MHz以上のユーザーにプレイさせるといじめられるので，気をつけたい。

バルログも「好きですアタック」を習得

9.95！　X68000版は新体操モードも装備(ウソ)

X68030では，(ちょっと仕様に問題ありの)多重AD PCMドライバがゲーム起動時に組み込まれ，効果音がポリフォニックになる。このフルソフトウェア処理による効果音の多重化はCPUパワーを浪費する処理なのであるが，馬鹿っ速いX68030には何の足枷にもならない。ゲームはどんなに効果音が重なろうが，背景動画が派手だろうがまったく快適に動いてしまう。

さて，これは各自の責任で行ってほしいのだが，X68030以外でも AUTOEXEC. BATの，

SET SP2CNF=00｜｜set SP2PCM=1

を，

SET SP2CNF=30｜｜set SP2PCM=4

とすれば，効果音が多重再生される。ただし，10MHz機ではすさまじいまでのスローモーション状態に陥り，まるで月面世界のよう。XVI系16MHzマシンでは，時折重くなるがまぁまぁ実用レベル。単音通常モードでの軽さを覚えてしまうとやや遅いと感じるかも。RED ZONE系(24MHz改造マシン)ではまったく重くならず快適。常時ポリフォニック効果音モードで遊びたい。

マシン性能に左右されるものの，プログラムが非常に頑張ってかなりのパフォーマンスを実現したということで，引き分け。

操作性対決

ストIIシリーズは，格闘ゲーム標準となってしまった8方向スティック＋6ボタンでのプレイを前提に作られている。X68000では8方向スティック＋2ボタン(またはキーボード)でもプレイできるが，これでは高度なアクション操作ができず，100%楽しめない。そこで，前作ストII'に付属していたアダプタを使用して6ボタンスティックを接続することになる。これは，先月号の「餓狼伝説SPECIAL」の記事で紹介した変換コネクタでもよい。ただし，ゲームオプションメニューで「JOYSTICK」の項目を「CPSF-MD」にし，さらに「JOYSTICK CONFIGURATION」で弱パンチから強キックまでの6ボタンの割り当てを「X，Y，Z，C，A，B」に設定する必要がある。

さて，この勝負，ユーザー各自が好きなジョイスティックを使用できる分，少し上かな，ってことでX68000版の勝ち。

終わりに

トーナメントモードについては，来月号をお楽しみに。また，ストII中毒ライター陣による各キャラ別攻略も始まる予定だ。

やっぱXにはXだよなぁ

スーパーストIIが移植されて，X68000ユーザーとしてはすごく嬉しいのだけれど，実際にプレイしてみるとなんか地味に感じられる。これは禁句かも知れないが，ゲーム好きのX68000ユーザーは皆そう感じているはずだ。我々は半年以上も前から最新作「スーパーストリートファイターII X」をプレイしているのだから。

「X」になって新しく採用されたスーパーコンボで，さらに煮詰まった各キャラの性能バランスと戦略性，ついに囲碁将棋に並ぶほどのゲーム性が獲得された感がある。「スーパー」は「X」という頂点を目指すうえでの単なる経過点に過ぎない。X68000版「スーパー」を「X」にするパワーアップパッケージを出しませんかね。いやホント，冗談抜きで。

総合評価	0　5　10
BGM	★★★★★★★★
サウンド	★★★★★★★★
グラフィック	★★★★★★★★★★
処理速度	★★★★★★★★★
操作性	★★★★★★★★★★

特別編1

ナンでもイケるイカシた奴

Terry Bogard

Nishikawa Zenji 西川 善司

「餓狼伝説」シリーズ出演中のテリー氏がタイトー「カイザーナックル」のバーツ氏を必殺技無断盗用の疑いで訴えました。テリー氏は氏のバーンナックルをバーツ氏がサンダーナックルとして無断盗用していると主張していますが，バーツ氏はこれについて全面否定をしています。しかし当局の強制捜索の結果，バーツ氏はテリー氏の必殺技クラックシュートにそっくりな技・ローリングスマッシュも隠し持っていたことが判明し，バーツ氏は不利な状況に置かれています。……臨時ニュースです。ただいま「ストリートファイターII」出演中のガイル氏が「餓狼伝説」シリーズ出演中のキム・カッファン氏を必殺技無断盗用で……。

2つの戦法

テリーは飛び道具を備え(パワーウェーブ)，さらに一瞬無敵の対空技(ライジングタックル)，加えて2つの突進系技(バーンナックル，クラックシュート)を持っている。いわば，待ってよし，攻めてよしのオールマイティキャラだ。

テリーを使っての基本的な闘い方は，パワーウェーブを撃って跳ばせてクラックシュートやライジングタックルで落とす，相手の挙動に対して的確に対処する受動的な攻め方と，通常技とバーンナックル，クラックシュートの高速コンビネーションで相手を一方的に畳み込む能動的な攻め方の2通りがある。

まず，受動的戦法について。

テリーの必殺技ライジングタックルは出かかりが無敵。地上からにしろ空中からジャンプで跳び込まれたにしろ，とにかくライジングタックルを出せればそれだけでいい。最悪でも相打ち。なんらかの攻撃で相手の跳び込み攻撃を誘発し，ライジングタックルで撃退するのだ。ライジングタックルは上方向だけでなく「ラリアット」のように伸ばした腕にも攻撃判定があることを覚えておこう。ノコノコ近寄ってきた敵を確実にブッ飛ばせる。また，密着状態で出せばゲシゲシと何度もラリアット部分が敵にかすり，絶好の削り技にもなる。

パワーウェーブは確かに敵に跳ばせるきっかけを作る重要な技だが，出すまでと出したあとのスキが大きいので敵との間合いの取り方は慎重に。撃退するときにライジングタックルが間にあわなければ，しゃがみ大キックや立ち小キックを出すといい。

次に，能動的戦法だ。

いつからか，技が出ているときに次の技を出すことを「キャンセル技」というようになった。同じく有名なゲーム隠語「めくり」と同様，ゲーメストあたりが広めた言葉だろうが，SPECIALでもこのキャンセル技がポンポン出る。もともとはバグ技的なモノだったのに，格闘ゲームでは必須事項のゲーム性となってしまった。

さて，テリーは大ぶりなしゃがみ大キックやしゃがみ大パンチその他の通常技を出したのち，先行入力のようなイメージで必殺技を出すとそのキャンセル技が簡単に出る。

能動的戦法とは，相手と近い間合いで，このキャンセル技テクを使って次々に技を敵に打ち込んでいく戦法なのだ。無数の技の連発に固められた敵がひるんで無駄な動きを見せれば，そこに技が命中する。命中しなくても戦況を自分有利に進められる。

ただ，やみくもにやっていると敵の一瞬無敵系必殺技にいとも簡単に撃退されるので，戦略的に用いよう。いかに間合を詰めるか，いかに敵を固めるかが鍵となる。これで敵に一発でも技が決まり，運がよければ連続技になる。この戦法は終盤の卑怯な体力削りにも最適だ。

ライジングタックルは上昇中は無敵

連続技

「餓狼伝説SPECIAL」の場合は，公式的な連続技はあるにはあるようだが，その場その場のアドリブで決まることが多い。やや接近状態で小パンチが数発入れば小クラックシュートや小バーンナックルへ簡単につながるし，うまい人なら小パンチ数回→大パンチキャンセル→小クラックシュートなどを出してくる。テリーに限らないが，連続技の鍵は小パンチにあり，といったところだ。慣れてきたら小パンチ数回→大通常技キャンセル→必殺技を練習してみよう。

パワーゲイザーの使い道

パワーゲイザーは，相手の転んだところへ重ねて出すという「削り」用途にしか使い道が見出せない。対空に使うのは，技が出るまで時間がかかるので，ものすごい先読みが必要。実戦にはほとんど使えないので，あてにしないほうがいいだろう。

まずはバーンナックルで

対空でパワーゲイザーが使えるとよいが……

特別編2

すべての技を尽くして戦え！ Andy Bogard

Taki Yasushi 瀧 康史

キャンセル技を使いこなせ

アンディに限らず「餓狼伝説SPECIAL」は，いかにうまく通常技をキャンセルさせ必殺技につなぐかで，プレイの美しさが変わる。もちろん，キャンセル技ができなくても，勝つことはできるだろうが，それでは面白さは半減すると思う。

キャンセル技をいまさら説明する必要はないだろう。要は通常技を出している最中に必殺技コマンドを入力し，通常技の動きを途中でキャンセルして，必殺技につなぐという一連の動作だ。最初の通常技が当たってから，技が戻るアニメーションをキャンセルして必殺技を出せば，連続技にもなり，与えるダメージが増すという仕組みだ。

ただし，キャンセル技＝連続技というわけではないので，そのあたりは試行錯誤してみてほしい。

表1はアンディのキャンセルできる通常技の一覧。○がついているものは，通常技が出ている最中に必殺技を出せばキャンセルできるというわけ。必殺技は，飛翔拳でもよいし，空破弾でもよい。つまりどんな必殺技でもよいのだ。

これ以外に，避け攻撃とライン飛ばしも，キャンセルができる（全キャラ共通）。

キャンセルできたからといって，連続技になっているわけではない。たとえば，立ち小パンチと立ち小キックは，キャンセルできても連続技にはならない。敵があわてて防御をすれば，技がつながらないのだ。

それ以外にも，大キック，避け攻撃は，必殺技につながっても，敵が転んでしまえばつながらないし，ライン飛ばしは当たってしまえば敵があちらのラインにいってしまうので意味がない。ただ，アンディ以外のキャラは，大キックが鋭利な場合が多いので，大キックキャンセル飛び道具で，牽制を兼ねて削れる。アンディならしゃがみ大パンチがよいかな。

キャンセルに対する知識はこんなところ。

シャケ攻撃は必ず出せるようになろう

キャンセルの超裂破弾

アンディを使うからには，超技をうまくキャンセルで出したい。超技の超裂破弾は，真下(↓)溜，↘，→＋弱，大キックボタン同時押しなので，いちばん簡単なのは，しゃがみ大パンチキャンセル超裂破弾。これは連続技。しかし，ダメージは実は近距離立ち大キックのほうが大きいわけで，できるならこれをキャンセルしたい。

通常，必殺技は，レバー入力後0.1〜0.4秒ぐらい，ボタン入力の猶予時間がある。そこで，これを利用して大キックをキャンセルすることができるのだ。

具体的には真下に普通に溜め，グルっとレバーを回して，大キックのみを押して，即座に弱と強を同時押しする。同時押しは連射でもいいかな。こうすれば，近距離強キックを見事キャンセルして，超裂破弾が出るのだ。

SPECIALでは空破弾が使える

この理論を応用すれば，大キックをキャンセルして大斬影拳，など多彩な攻めができるので，ぜひマスターしてほしい。難しいだろうけど。

その他

最後に私の「癖」のようなものを書こう。

まず，しゃがみ技だが，アンディはしゃがみ大キックよりも大パンチのほうがリーチが長い。そのくせ，くらい判定が小さいので（つまり強い技？），変なタイミングでしゃがみ大パンチをちょびちょび出すアンディは割といやらしい。

そして対空技だが，昇龍弾が意外と使えない。「無敵である」ということ以外に利点がない。その理由はまずダメージ。多く使われるだろう弱の昇龍弾よりも，実は，近距離の立ち大キックのほうが大きい。そしてこの立ち大キックは，くらい判定も小さく，一方的に攻撃が当たる。おまけに遠めでは多少ダメージが小さいものの，斜めに出るので，危険性も少なく，対空にかなり使えるというわけだ。

アンディというキャラは，SPECIALではもはや圧倒的な強さは持ち合わせていないが，それでもうまく使えばかなり強い。まさに，すべての技を尽くして戦え！　だ。

表1　キャンセルできる通常技一覧

	しゃがみ	立(近)	立(遠)
弱パンチ	○	○	○
強パンチ	○	○	×
弱キック	○	○	○
強キック	○	○	×

対空技では強キックを使うほうがよい

リョウ・サカザキも，やはり金的は痛いか？

特別編3

キムチの国のテコンドー達人

Kim Kap Hwan

Asakura Yuji 朝倉 祐二

餓狼伝説2では使い手に技術を要求したキム・カッファンだが，SPECIALになって誰でも使える基本パフォーマンスの高いキャラクターに変身して帰ってきた。前作からのキム使いにとっては喜ぶべきことなのだが，手放しで喜んでいるわけにもいかないようだ。というのは，真面目人間で悪が嫌いというキムが，基本技，連続技のあまりの強さゆえ，いつしかほかのキャラ使いに「極悪キム」と呼ばれるようになったからだ。キム使いの心境は複雑といえよう。

連続技がキムを強くした！

キムを使ううえで絶対に覚えておきたいこと。それはパンチ，キックからつなぐ小半月斬攻撃である。SPECIALになってからというもの，キムは通常技のほとんどをキャンセルして連続技を入れられるようになった。だからパンチ，キックを出したあとにキャンセル小半月斬を入れることにする。相手がこちらの攻撃をガードしたとしても，そのあとに必ず小半月斬をくらってくれるのだ。地道だが，体力を削るにはうってつけの戦法である。

そういえば前作の餓狼伝説2ではライン飛ばし攻撃をキャンセルしての大半月斬が基本技だったが，相変わらずライン飛ばし攻撃はリーチが長いからこれも使える。織り交ぜて使ってみよう。

飛翔脚，飛燕斬，鳳凰脚はどうなる？

キムはほかに飛翔脚，飛燕斬といった必殺技を持つが，これらは前作まで対人間の対戦で使われることの少ない必殺技だった。というのは，ガードされたあとの隙が大きかったりするからだ。ところがSPECIALでは，これらの必殺技が対戦で使えるようにすべて動作が変更になったのだ！

まず飛燕斬を出した直後に無敵時間ができた。自分がダウンしたところに相手が近づいてきたら，起き上がりに飛燕斬を入力しておくことだ。必ず相手に当てることができるぞ。

もう一方の飛翔脚だが，テリーのパワーウェーブ，チンの氣雷砲など相手が飛び道具を出した直後を狙って出すと，綺麗に決まるのは当たり前なので，もう一歩進んで移動手段としての飛翔脚を知ってると，さらに強いキム使いになれる。つまり相手に当てない飛翔脚を出して間合いを詰め，そこから基本技で攻撃するのだ。ジャンプしたらすぐに大飛翔脚を出せるように練習しよう。

さて，キムの超必殺技である鳳凰脚だが，ほかの例に洩れず，扱いやすくなっている。前作でガードされたときは，超必殺技であるにもかかわらず体力を削ることすらできなかったのだが，SPECIALでは体力を削ることができるようになった。これは大きな進歩である。特に対戦相手がベアならば，鳳凰脚を連続で出すと敵は何もできずに大地に横たわっていることになるだろう。

であるが，鳳凰脚も完全無敵というわけではない。キムの鳳凰脚はリョウ・サカザキの竜虎乱舞と同じように，鳳凰脚コマンドが入力されてから，相手との間合を詰めて接近していく間にも小キック一発で返されてしまうこともあるのだ。これに対しては，なんらかの通常技をキャンセルさせることによって鳳凰脚が入れられる。練習が必要とはいえ，頼もしいことだ。

こんなことをやられたら屈辱だ

連続技は悪か!?

たしかにキムの連続技は強烈だ。小パンチが当たれば，そこからキャンセル鳳凰脚も入るし……。連続技は悪とは思わないが，「極悪キム」の名が高まらないうちに，たまには連続技に頼らない試合運びをやってみたらどうだろう？　どんなに攻撃力が強くても，ワンパターンな攻撃では相手に読まれてしまうし，それこそハメてるつもりはなくてもお決まりのルーチンワークで勝ち進んでいれば，相手に不快感を与えることになるだろうからね。たとえば小パンチキャンセル半月斬だが，相手は小パンチが当たった時点で半月斬が来ると思ってガードしてるだろう。そこをキャンセルせずに，相手が「おや？」と思って攻撃してくるところを，半月斬を一瞬遅らせてかますとか……。こっちのほうが，きたね～って感じかなぁ。

聞いたところによると，跳び込み大キック→大パンチ2発→キャンセル鳳凰脚で一発KOできるというので，これを決めることがキム使いの目指す目標となるだろう。地道に敵の体力を削るよりも，やっぱり最後は鳳凰脚で決めたいもんね!

飛燕斬は上昇中は無敵だ

今日から私が教育してやる

特別編4

ヨ〜グ〜ルトいっぱい食べ〜る

Wolfgang Krauser

Sudou Yoshimasa 須藤 芳政

今回はヴォルフガング・クラウザー君の紹介です。

ボール紙で自作したようなチャンピオンベルト。弁慶の泣きどころを覆った大げさなプロテクター。ラーメンを食べるときに邪魔なヒゲと長髪。２メートルの長身が躍動するさまは，まさにジャイアント馬場ターボ。ヴォルフガングという名も，アマデウスをパクったのか実際に関係があるのか怪しい。どうですか？　実に魅力的なキャラクターです。

「クラウザーが闇の帝王なら，僕の父さんはキャバクラの帝王だい！」という方も試しに使ってみてください。豪快な殴りと蹴りの末に勝利を手にし，「ヨ〜グ〜ルト，いっぱい食べ〜る」を耳にしたら最後。貴方はクラちゃんに首ったけです。なぜヨーグルトをいっぱい食べるのかは，私も詳しくはわかりませんが，ヨーグルトが好きなんでしょう，きっと。

ふあ〜！ ふあ〜！

この男は，とにかく格闘中に「ふあ〜！ふあ〜！」とうるさい。「レッグトマホーク！」「カイザーウェーブ！」ロボットもの

K「スキンシップはいいよなぁ」

アニメじゃないんですから，いちいち技の名前をいわなくても出ます出します出させますと私はいいたい。

しかし，通常技の攻撃が強力で相打ちになってもこちらのほうが絶対的に有利！パンチやキックの決まる間合が広い！　上下に撃ち分けられる飛び道具！　これだけ揃っていれば，ほかのキャラが新宿中央公園ならクラウザーは新宿御苑といえるでしょう。

そして中段当て身投げ！　これを決めることのできるガラパゴス島在住のお爺ちゃんお婆ちゃんはいないといわれるほど難しい技なので，決めたときには自分が人間として１まわり成長したような気がします。失敗するとラジオ体操をしているように見えてしまうのがショーック！　これから日本の将来を背負ってゆく身なら，確実に決めたいですね。ただし，ギースの上段当て身投げは許せません！　あのような，テレビを見ながらエアコンかけてお菓子を食べながら決められる技(じゃ〜ん！　電気を大切にね)……かどうかはわかりませんが，とにかくあれは「決められすぎてムカツク」

A「はい！　ボス」

という理由で許せません。

通常の闘いにおいては，近距離でケズるなら，立ち弱キックキャンセル弱レッグトマホークまたは下ブリッツボールでいじめます。相手のすぐ横で立ち弱キックを出すと，ザンギエフの立ち小キックのような蹴りが出てとても愉快。立ち強パンチからの弱レッグトマホークは結構強力なので，跳び込みからの３段を狙いましょう。

カイザー！ ふあー！

ライフゲージが点滅し始めるとカイザーウェーブを決めたくなるのは，人間として当たり前の欲求です。しかし，単にずりずり後退していたのでは相手に「へっへ，こいつ溜めてるぜ」と感づかれて足払いや飛び込みを食らい，「カイザー！」まで口走ったところで「ふぁー！　ドスンドスン！」とブッ飛ばされるのがオチです。カイザーウェーブは，相手の起き上がりに重ねるかブリッツボールを撃った直後に溜め始めるなどの工夫が必要です。それ以上に，普通に闘ったほうが勝てる気もしますが，カイザーウェーブを使わなくなったクラウザー使いは，まるでホーホケキョと鳴かない江戸屋猫八のような味気なさを感じます。

デカい奴は強いのだ！

昔から図体のデカい奴は強いと相場が決まっている。口から吐き出すもののような名前の爺さんや，ヒヨコの飼育係なんてメじゃないぞー！　レディー男はタダの脂肪の塊だー！　風船男など論外だぞー！　というわけでクラウザーは最強なのです。

さてと，キムを使うとするか…………。

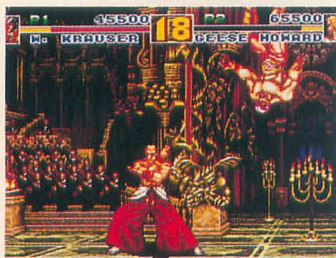
K「悪いのはどの手だぁ」

K「これがカイザーウェーブです」J「うわー」

J「お前のかーちゃんデーベーソー」

1994年10月号のハガキ集計ベスト10 最近買って気に入ったソフトは？

POINT	タイトル	発売元	発売日
114	餓狼伝説SPECIAL	魔法株式会社	'94/7/28
72	スターラスター	電波新聞社	'94/8/26
54	SX-WINDOW ver.3.1	シャープ	'94/5/30
20	ストリートファイターⅡダッシュ	カプコン	'93/11/26
19	悪魔城ドラキュラ	コナミ	'93/7/23
17	クイーン・オブ・デュエリスト外伝α+	TAKERU	'94/8/10
12	ぷよぷよ	SPS	'94/3/25
10	スーパーリアル麻雀PⅣ	ビング	'94/4/27
8	ジオグラフシール	エグザクト	'94/3/15
7	レッスルエンジェルス3	TAKERU	'94/7/31
7	Mr.Do！/Mr.Do！vs UNICORNS	電波新聞社	'94/7/2
7	あすか120% BURNING Fest.	ファミリーソフト	'94/4/22

（無作為抽出した1000通のハガキを集計）

全体の印象は，先月号の集計とあまり変わりないようですが，2位に8月末に発売された「スターラスター」が入りました。

この「スターラスター」はアンソロジーシリーズではありませんが，懐かしの名作という点では同じような位置づけのゲームで，発売を心待ちにしていた人が多かったようです。店頭では，なぜかアンソロジーシリーズと同じ大きさのパッケージを探してしまい，見つけるまでに時間がかかったなんて人もいるようですが……。忠実移植のアンソロジーだけではなく，このような企画に対しては喜びの声も多いので，メーカーさんにはこれからも充実したラインナップを図ってほしいものですね。

あとは，変化といえば，固定ファンによる着実な得票を続けて，今月トップ10に返り咲いたのが7位の「ぷよぷよ」と8位の「スーパーリアル麻雀PⅣ」です。これらは，ゲームの面白さや完成度に加え，発売から数カ月を経ながらも同じジャンルのゲームがその後発売されないという状況でもありますので，今後も変わらず支持され続けるのではないでしょうか。

さて，全体の得票数を見ると，上位陣がそれぞれ安定しているのに比べ，10位前後のものになると人気度も横並びとなっています。このあたりは，1票増えただけでもすぐに順位が入れ替わってしまいますので，特にお気に入りとか，ほかの人におススメしたいとかいうソフトがある人は，必ず忘れずに，早めにアンケートはがきを出してくださいね。

はがきといえば，最近はソフトに対するコメントを書いてくれる人がやや減りつつあります。「当然コレ！」なんてのは，気持ちはよくわかるのですが，編集部としてはレビュー記事への感想も含めて，そのソフトの感想などをいただけると嬉しいと思っています。もちろん，期待はずれだったものについての意見などでもかまいませんので，いろいろな声を聞かせてください。お待ちしています。

来月号ではいよいよ「スーパーストリートファイターⅡ」の登場です。果たして新記録が生まれるか，得票数に注目です。

【特集】

STEP UP BASIC

誰もが一度は触れてみる言語BASIC。
初心者用のプログラミング言語と侮られがちな面もありますが,思っているより多くのことがBASICでできるものです。
プログラミングを知らない人やできないと思い込んでいる人は,まずBASICに触ってみることから始めてみるべきでしょう。なんらかのプログラミング経験があるのとないのとでは「パソコン側の事情」に対する理解のしかたが違ってきます。そしてそれはなにかのトラブルが起こったりしたときに,どう対応すればよいのかの指標になってくるのです。
BASICでプログラミングできる人には,より大きな可能性があります。外部関数が増えることでX-BASICはその能力を何倍にも拡大するのですから。
ではBASICを再確認することから始めてみましょう。

CONTENTS

プログラミング入門心得

BASIC修得への道

Tamura Kento 田村 健人

BASICに限らず，プログラミングを学ぶときには一定の手順があります
プログラミングの感覚を身につけるためにはどうするのか
BASICへの基本的なアプローチ法を見てみましょう

本誌1994年6月号のアンケート分析によると，「BASICを使っている・理解できる」が全体の8割，「BASICに関心がある」は1割，「その他」が1割。とにかく，本誌読者の2割はBASICでのプログラミングを未経験なわけですね。短絡的に「BASICを未経験＝プログラムを組んだことがない」と考えれば，2割もの人たちはプログラミングの楽しみを知らずにいるわけです。

これはもったいないことです。

ゲームをやるのは楽しい，パソコン通信も楽しい，そしてプログラミングもとても面白いものなのです。パソコンを持っているなら，ゲームも通信もプログラミングも体験してほしいものです。

この記事では私の経験に基づいて，BASICを覚えるまで・覚えてからのコツを書いてみます。BASICをまったく知らない超初心者は最初から，ある程度使っている人は途中から読んでください。

プログラムってなんだ?

よくいわれるのが「コンピュータ，ソフトがなければただの箱」という言葉です[*1]。コンピュータは，ソフトウェア（つまり，プログラム）を与えてやらなければなにもしてくれません。

コンピュータにやってほしいことが順番に書かれているものが，プログラムです。プログラムをBASICで書くか，機械語で書くかの違いは，指令の細かさと安全性にあります。

たとえば，「風呂に入るプログラム」を考えます。

1．服を脱ぐ
2．風呂場へ入る
3．頭を洗う
4．体を洗う
5．湯船につかる

と命令するのがBASICで，

1．シャツの1番上のボタンを外す
2．シャツの2番めのボタンを外す
⋮
n．シャツを脱ぐ
n＋1．ズボンのファスナーを下ろす
n＋2．ズボンを脱ぐ
⋮

と，こと細かく命令してあげなければならないのが機械語だと思ってください。

上記の例で「服を脱ぐ」が2回現れた場合，BASICならば2回目の「服を脱ぐ」で「すでに服を脱いでいる」ということを検出してくれるでしょう。ところが機械語で書かれたプログラムではそういった矛盾のチェックは行われません。ボタンが6個しかないシャツを着ているのに「7番目のボタンを外す」という命令があった場合，存在しない7番目のボタンを外すことができずにそのまま止まってしまうでしょう。もしかしたら，7番目のボタンがあるあたりの皮膚をむしり取って，自分自身に致命的なダメージを与えるかもしれません。

機械語のプログラムは矛盾のチェックがないぶん速く実行されますが，プログラムに間違いは許されません。

コンピュータを理解するには機械語でプログラムを書けるようになるのがいちばんですが，やはり慣れないうちは安全なBASICを使うのがよいようです。

*1：　ディスプレイ・ハードディスク・モデムもなければ，やっぱりただの箱です。

プログラムを組めるようになりたい!

なにもわからない状態の人がなにをすればいいのかを書いてみます。X-BASICのマニュアルはわりと親切なのでそちらも見てください。

1）　使い方・打ち込み方を覚える

X-BASICの起動の方法，プログラムを打ち込む方法，打ち込んだプログラムの保存の方法，保存したプログラムの復帰の方法，プログラムの実行の方法，を覚えます。マニュアルを読むのがいちばんですが，簡単に書いておきます。

・X-BASIC起動

A＞

となっているところで，

A＞basic＜改行＞

と打つ。

・打ち込み

1行を打ち込んだら，改行キーを押すことが重要。

・保存

save　”名前”＜改行＞

・保存したものの復帰

load　”名前”＜改行＞

・実行

プログラムをすべて打ち込み終ったか，loadをしたあと，

run＜改行＞

2）読む・打ち込む

BASICのプログラムリストをひたすら打ち込みます。これはキーを速く打つ練習にもなるので，絶対に無駄にはなりません。わけがわからなくてもいいから，とにかく書いてある文字を打ち込むのです。「習うより慣れろ」という諺がよく当てはまります。

数をこなしてくると，「よく出るフレーズ」を覚えてしまうでしょう。

```
if ... then ...
for i=0 to ... next
while ... endwhile
func ... endfunc
```

これらのフレーズを見出せないなら，まだ打ち込みが足りません。しばらく打ち込み三昧の生活を続けてください。とにかく，「for」ときたら自然に「i」キーに手を伸ばしてしまうようにならなければいけません。

じゃあ，これらのフレーズを覚えたらどうするのか？　まず，マニュアルでこれらの意味を調べてみます。細かいことはわか

らなくても，おぼろげに役割がわかるでしょう。ちなみに，「for　i=0 to　N　……nextはN+1回の繰り返しをするものだ」と思って構いません。

試しに，

```
10 int i
20 for i=0 to 9
30     print "あん"
40 next
50 end
```

というプログラムを打ち込んで実行してみると，「あん」と10回表示されるはずです。本当はiをaにしても，「0 to 9」を「1 to 10」としても構わないのですが，せっかくコンピュータを扱うのですから，「数を数えるときは0から」や「スリーサイズをいうときは16進数で」は基本事項として心がけてください。通産省もforにはiを使うことを推奨しています*2。

まがりなりにも意味がわかったところで，また打ち込みに戻ります。しかし今度は「文字を打ち込む」のではなく「文を打ち込む」という気持ちでいてください。文字を読むのではなくプログラムを読むようにして打ち込んでいるうちに，プログラムの書き方が自然に身についていきます。

3）プログラムの改造

たとえばゲームのプログラムを打ち込んで遊んでみたとします。しかし難しいゲームだった。マイキャラが3機ではとてもクリアできないな，という場合，プログラムを書き換えてマイキャラを増やしてしまいましょう。これぐらいの改造なら，プログラム全体の流れを理解できなくてもなんとかなります。プログラムのなかで，

なんとか=3

となっている部分を見つけたら，'3'を'10'とかに書き換えてみます。それでrunしてみます。意図した通りの変化がなかった場合，書き換えたところを元に戻して，また「なんとか=3」を探して書き換えます。これの繰り返しで，いつかは思惑通りの結果になるでしょう。ただ，プログラムを実行していて見える数が'3'だからといって，プログラム中でも'3'であるとは限りません。書き方の都合で'2'や'4'になることもしばしばあります。

変数の書き換えよりは，キャラクタパターン・音楽・効果音の差し替えのほうが楽かもしれません。これらは定義する関数が決まっているので，どこを書き換えればよいのかがわかりやすいのです。パレットの設定も書き換えやすいでしょう。

とにかくいろいろなものを書き換えてみて，コンピュータが自分の思い通りに動いてくれることを実感してください。

＊2：ウソです。

プログラムを作ろう

まったく0の状態からプログラムを作り始めるときに，どこから手をつけるのかという手法でトップダウンとかボトムアップとかいうのがありますが，とりあえずは無視します。重要なのは，いつでも動かせる状態でプログラムを組むことです。まずは，ジョイスティックに従って文字'A'が動くプログラムを作ります。ちゃんと動くようになったら，'A'という文字をスプライトにしてみます。これもちゃんと動くようになったら，壁を置いてみる……などなど。

かなり複雑な処理を書くときは，一度紙にメモなどを書いて頭を整理してからプログラムを書くようにします。ただ，フローチャートはできる限り書かないほうがよいでしょう。フローチャートは逐次的な処理の記述はわかりやすくなりますが，繰り返し構造などがとても見づらくなります。よって，普通のBASICやアセンブラなどには役に立つのですが，X-BASICのような普通でないBASICやC言語ではフローチャートからプログラムを作るのに余計に手間がかかります。

「昨日の自分は今日の他人」ありがたいお言葉です。これを心にとどめて，わかりやすいプログラムを書くように心がけてください。「自分しか見ないプログラムなんだから，コメントなんていらないや」これは大間違いです。自分の記憶力を信用してはいけません。コメントは自分のためにつけるのです。

プログラムをわかりやすくするために，変数名や関数名を日本語で書くという手は残念ながら使えません。X-BASICもXCコンパイラも日本語の変数名，関数名は扱えないのですね（アセンブラは使えるのに……）。SX-BASICでも使えませんが，これはきっと石上さんに頼めば使えるようにしてくれるでしょう。

慣れないうちは「このフレーズは二度出てくるからfuncにすればプログラムが短くなる」というぐらいの認識でも構いませんが，慣れたらそういう考えは捨てます。「独立して意味のあるもの」をfuncにするのが正解です。一度きりしか呼ばれないものでも，意味のある塊ならばfuncにしたほうが見通しがよくなります。たとえば，

```
10 int .../*  変数の宣言
 :
50 Initialize()/*  初期化
60 Mainloop()/*  本処理
70 Terminate()/*  後片付け
80 end
90 func Initialize()
100    InitializeChar()
110    InitializeMap()
120    InitializeMisc()
130 endfunc
```

a*16よりa shl 4のほうが速い？

どうしてa*16とa shl 4が比較できるのかということも，わかる人には当たり前ですがわからない人にはわからないですね。

整数では，a*16とa shl 4は同じ値を表します。なぜ同じになるのかは，両手の10本の指で1024まで数えられるようになってから考えてください。

なんで速くなるのかは話が長くなるので省略して，どれぐらい速くなるのかを見てみます。

リスト1は「i*16」「i shl 4」「i/8」「i shr 3」「i*20」「(i shl 4)+(i shl 2)」を50万回ずつ計算するプログラムです。

うちの環境での実行結果は以下のようになります。

1	i*16	209秒
2	i shl 4	190秒
3	i/8	206秒
4	i shr 3	190秒
5	i*20	209秒
6	(i shl 4)+(i shl 2)	309秒

1と2,3と4,5と6がそれぞれ同じ値の計算です。ぱっと見てわかるとおり，shl/shrを使ったほうが高速に動いています。ここでは9％ほどの向上ですが，forループのぶんを考えるとそれ以上の効果はあるはずです。6が遅いのは式の解釈に時間がかかるためでしょう。

リスト1

```
 10 int i,a
 20 print time$
 30 for i=0 to 499999
 40     a = i * 16
 50 next
 60 print time$
 70 for i=0 to 499999
 80     a = i shl 4
 90 next
100 print time$
110 for i=0 to 499999
120     a = i / 8
130 next
140 print time$
150 for i=0 to 499999
160     a = i shr 3
170 next
180 print time$
190 for i=0 to 499999
200     a = i * 20
210 next
220 print time$
230 for i=0 to 499999
240     a = (i shl 4)+(i shl 2)
250 next
260 print time$
270 end
```

140 func ...
とするぐらいの心意気がほしいところです。

BASICからの離脱

やっぱりBASICというのは遅いものなので，プログラムを速くするにはどうするかということを考えます。N88-BASICだと「DEFINT A-Z」とか「変数名は2文字以内」とか「'よりREMのほうが速い」などの姑息な高速化手法がたくさんありますが，X-BASICではそういったテクニックは聞いたことがありません。もちろん，コンピューター一般にいえる「a＊16よりa shl 4のほうが速い」，「a/8よりa shr 3のほうが速い」というのはあるでしょう。

BASIC上での高速化手法が確立されていないのはBAStoCの存在によるところが大きいと思います。どんなにタコなプログラムでも，BAStoCをかけることによって劇的な速度向上が望めます。設定さえきちんとしていれば，

A＞cc hogehoge.bas

とするだけで簡単にhogehoge.xができます。

人間の欲望は限りがないもので，そのうちBAStoCをかけても遅いと感じるようになるでしょう。最後の手段として，BAStoCにgccを使うというのがあります。これで2倍程度の高速化ができるようです。gccはXCコンパイラに較べて設定する項目が多くて大変かもしれませんが，それに見あうだけの結果にはなります。gccを使う場合でも，ただ単に，

A＞cc -B hogehoge.bas
A＞gcc hogehoge.c -lbas -lfloatfnc

とするだけではあんまりおいしくありません。最低でも，

A＞gcc -O hogehoge.c -lbas -lfloatfnc

できれば，

A＞gcc -O -fomit-frame-pointer -fforce-mem -fforce-addr -fstrength-reduce hogehoge.c -lbas -lfloatfnc

ぐらいのオプションを指定してください。でないと，gccの真価は発揮されません。

BAStoCも含め，X-BASICをマスターしたかどうかは，以下のプログラムで判定しましょう。

d＝stick(1)
x＝x－((d＝4) and 1)＋((d＝6) and 1)
y＝y－((d＝8) and 1)＋((d＝2) and 1)

このプログラムがよく考えられて書かれていることが理解できるでしょうか？ なにをしているのかまったくわからない場合は，まだまだBASICの修行が足りません。「and 1はなんのためにあるんだろう」と思った人はBAStoCを使いこなしていません。

次のステップ

ありがちな進路として，BASIC修得後にはアセンブラを学ぶかC言語を学ぶかになります。ここで，X-BASICとC言語の見た目の類似性や，BAStoCで対応関係がわかりやすいなどの理由でC言語に進みやすいようですが，はっきりいえばそれは間違いです。アセンブラを先に修得してください。アセンブラをまったく経験していない人の書いたCのプログラムは，どうも洗練されていません。

時代の趨勢からいって最終的にC言語にたどり着いていくものなのかもしれませんが，ならばなおさら，C言語をよりよく学ぶためにもアセンブラの学習は欠かせません。BASIC,アセンブラ，C言語の順でプログラムを極めていってください。

10月号の付録ディスクの不具合

特集とはなんの関係もありませんが，先月号のシャーペン外部コマンドとarlkのバグについて報告します。

●xclick

先月号編集後記に書いたように，これを組み込んでいる状態で「マルチビュー」→「シングルビュー」とするとシステムエラーになります。

条件によってはトリプルクリックしないとデータ付きイメージに反応しません。とりあえず，イメージの後ろにスペースを入れることでダブルクリックで反応するようになります。

●setkind1

パラメータを実行時に入力し，もう1回パラメータを入力すると，前回の入力と混ざってしまいます。

●optab，ecbrace,rpar

疑似ダイアログがあるときに実行したとき，疑似ダイアログを強制的に閉じてしまいます。

各ソースの，main()関数中のdisposecomm();の行を削除してください。

●isearch

CTRL＋Gで終了すると，再び実行できなくなります。

「¥」などを検索すると，2バイト文字の2バイト目にマッチしてしまい，なにも表示されません。また，外部コマンドmapで割り当てたキーが入力できない件も修正しましたが，変更点が多いので割愛させていただきます。

●arlk

-ac -xc -ec のとき小さいデータを標準入出力に流すと，0D0Aと0Aの変換が行われてしまいます。あいにくソースが収録されていませんので，-ac -xc -ec は使わないようにしてください。-oで代用してください。

＊　＊　＊

たいへんご迷惑をおかけしました。ごめんなさい。

```
xclick.c ================
117:        if ( p >= q )  return 0;
118:        off = p - *(*tehdl1)->text-2;
          ↓
117:        if ( p > q ) {
                clicktime = _val->erec.ts.when;
                return 0;
            }
118:        off = p - *(*tehdl1)->text-2;
----------------
221:    return ret;
         ↓
221:    return 0;
----------------
setkind1.c ===============
 75:    if ( l <= 0 )  return ret;
 76:    ps = (short*)(buf+700);
         ↓
 75:    if ( l <= 0 )  return ret;
        buf[l] = '¥0';
 76:    ps = (short*)(buf+700);
----------------
isearch.c ===============
161:        MMPtrDispose( pgv );
162:        break;
         ↓
161:        MMPtrDispose( pgv );
            _val->excon[ide] = 0;
162:        break;
----------------
```

アルゴリズムを具体化しよう

コーディングの流れを見る

Nakano Shuichi 中野 修一

ひとつの処理を実現するための方法は実にたくさんあります
その過程でどのような思考が行われて命令が組み合わされていくのか
プログラミングの具体例を通して見てみましょう

プログラムを作るときには，使用する言語の種類というのはたいした問題にはなりません。速度と言語仕様（特にデータ型）で向き不向きはありますが，X68000で一般的に使われるX-BASIC，C言語，アセンブラでは，コンパイルなどで最終的にマシン語の状態で実行可能ですから速度の差は致命的ではなくなります。データ型などは基本的なものがあればなんとかなるものです。どの言語処理系を選ぶかという際には，結局のところ好みに依存する部分がかなり大きくなってきます。

で，やらなければならないことはどれも大差ありません。どれを選んでもいいのですが，我々にとってもっとも手近なのがBASICです。

つまるところ，BASICでのプログラミングというのはドミノ倒しのようにあらかじめいろいろな仕掛けを用意しておいて，配置が終わった時点でおもむろに実行するのと似ています。まあ，プログラミングとはたいていそういうものなのですが，BASICの場合はさらにそれが小刻みにやりやすいという特性があります（残念ながらSX-BASICはそういう環境ではなくなってしまいましたが）。

この小刻みにトライ&エラーが繰り返せるという点がBASICの最大のメリットです。

いきなり大きなプログラムを作れる人というのはそうはいません。マシン語で書かれた高度なプログラムも，開発の最初はX-BASICのテストプログラムだったりすることも少なくありませんし，最終的にものすごいプログラムができあがっていても，それはたいてい徐々に拡張されていったものです。根本的な改良とは違い，すでに動いているプログラムをベースにして機能を積み重ねていくことはそれほど難しいことではありません。

プログラムは積み重ねです。誰でも1歩1歩進めていかなければなりません。経験を積めばそれだけ一度に作れる部分が大きくなっていくのです。

*　*　*

プログラム作成にはいくつかの勘所があります。それはいくつかプログラムを作り上げたり，他人のプログラムを打ち込んでいるうちに身についていくものです。

で，問題は本誌に掲載されているプログラムなのですが「リストがすべてを語る」とはいうものの，掲載されたリストでは部分部分が断りもなく最適化されていたり，そのコードに至った過程というものがまったく省略されることが多々あります。

私もX-BASICのプログラムを組むことがありますが，どうも理屈の部分で込み入ったプログラムを作ることが多いのでコーディングについては説明していないことも多くて反省しています。以下では，簡単なツールを作る過程をできるだけ細かく解説してみたいと思います。

なにを作るべきか

プログラムを作れない人の原因の多くは，作成すべきプログラムというのが明確ではないことでしょうか。

プログラムを作成するにはなんらかの目的が必要です。プログラムする題材が決まった時点ですでに必要な処理や仕様の半分くらいは決まってしまいます。

プログラムを作りかけることのできる人と一応完結したプログラムを作成できる人の差の半分は題材を明確にできるかどうかのような気がします。あとの半分は，プログラムをうまくまとめて，うまく割り切ることができるかどうかです。ちょっとしたプログラム程度なら，特殊な才能や，コツコツとした作業が得意な性格（私はかなり違う）などは要求されませんので安心してください。

ということで，今回選んだ題材ですが，一部のスタッフからの辞書メンテが使いにくいというリクエストにお応えして，辞書をテキストにダンプしたものを加工するためのツールを作ってみることにしました。まあ，ASK3で辞書のメンテナンスをやろうという人がどれくらいいるかはわかりませんが……。

大まかな方向が決まったところで，さらになにをやるかを限定していきましょう。考えてみると辞書のメンテナンスで実際に必要になるツールというのはそう多くはありません。多くは標準コマンドで足りたり，辞書メンテに任せたりできるからです。

「テキストファイルをいじる，といっても，必要になりそうな処理＝キーワードごとに取り出したり単純にソートしたりすることは標準コマンドで簡単に処理できます。が，もうちょっと込み入った操作になるとお手上げです。ここでは，ダンプされたテキストファイルのマージ（結合）を行ってみましょう。

ちなみにどうしてこういうものが必要かというと，辞書メンテで「辞書マージ」などを行うと接頭語などの学習コードがおかしくなっていくので，

ダンプ→データ変換→ダンプ……

といった繰り返しができないからです。一度ダンプしたらあとはテキストファイル上でのみ編集を行うのがいちばん安全なのです。

変換に使用する辞書メンテでは，登録語の読みが順に並んでいないとまずいので，単にテキストファイルをくっつけただけではいけません。テキストファイルを結合してSORT.Xを使うという手もありますが，これだと故意に変更した候補配列が破壊されるのであまり望ましくはないでしょう。

ということで，やはりなにかツールを作るのがよいことになります。用途を見ても学習内容の違う辞書からデータを持ってきたり，品詞別に分離した辞書をいじってい

るときにはこのようなツールがあると便利です。

マージする

まず，この処理の前提として，2つの辞書ダンプファイルが必要になります。両方ともソートの必要はないもの，と仮定しましょう。こういう条件なら両方のファイルの先頭から処理をしていけばよいだけなので比較的簡単なプログラムですみそうです。処理の概要は，

1) 先頭のデータを取り出す
2) 比較する
3) 小さいほうを出力
4) 出力したほうに次のデータを補填
5) 2)へ

というのを繰り返します。

辞書の構造を見てみましょう。表1を見てください。これは編集部で使われている辞書をダンプして，先頭部分を抜き出したものです。

単語の並びは基本的にアイウエオ順で，音引き（ー）はアの前，濁音などは通常音の後ろ，ここでは出ていませんが拗促音（ァ，ョなど）は通常音の前になります。表2または取扱説明書の漢字コード表を参照してください。単語の並び順は漢字コードの順と同じであることを確認できるでしょう。

表1　辞書ダンプの例

```
ア,亜,単漢字,40
ア,唖,単漢字,40
ア,娃,単漢字,40
ア,阿,単漢字,40
ア,会,ワ五,09001
ア,合,ワ五,09
ア,逢,ワ五,09
ア,あ,ラ五,08
ア,あ,ワ五,09002
ア,有,ラ五,08
ア,開,カ五,01
ア,明,カ五,01
ア,在,ラ五,08
ア,空,カ五,01
ア,編,マ五,07
ア,飽,カ五,01
ア,遭,ワ五,09
ア,亜,接頭語,4500000
アーカイバ,アーカイバ,名,24
アーカイブ,アーカイブ,サ変名詞,22
アーキテクチャ,アーキテクチャ,名,24
アーキテクト,アーキテクト,名,24
アーギュメント,アーギュメント,名,24
アーク,アーク,名,24
アーケード,アーケード,名,24
アーサー,アーサー,人名（名）,28
アース,アース,サ変名詞,22
アーチ,アーチ,名,24
アーチェリー,アーチェリー,名,24
アーティスティック,アーティスティック,名,24
アーティスト,アーティスト,名,24
アート,アート,名,24
アーバン,アーバン,形動名詞,20
アービトレーション,アービトレーション,名,24
アーマー,アーマー,名,24
アーマード,アーマード,名,24
アーミー,アーミー,名,24
アーム,アーム,名,24
アーメン,アーメン,名,24
アーモンド,アーモンド,名,24
アーリー,アーリー,名,24
アーリア,アーリア,名,24
アーリマン,アーリマン,名,24
アール,アール,名,24
```

●変数を宣言する

X-BASICで扱うデータは，処理時には変数に格納しなければなりません。そこで変数宣言が要求されてくるのですが，難しいことは考えず，数値なら単純にint，文字列ならstrとしておけばまあ間違いはありません。あえてfloatやcharが必要になる局面はそう多くはないものです。

X-BASICでは変数定義時に文字列の長さを指定しないと，文字列型変数は暗黙のうちに最大32バイト長のものとして扱われます。32バイトでもたいていの場合は足りるのですが，辞書の内容を見るとあきらかに32バイトを超えるものもあります。知らずに処理すると発見しにくいバグの原因にもなります。気をつけてください。

で，この場合はどの程度の長さを確保しておけばいいのでしょうか。ここではとりあえず，かなり長い単語として知られる「スーパーカリフラジリスティックエクスピアリドーシャス」が登録できるように128バイトとしました。普通の単語を扱うにはこれで十分でしょう。

●ブロックを抜き出す

辞書ダンプファイルは1単語で1行なのでfreads()でデータを読み込みます。

比較は，それぞれの「読み」の部分だけで行います。これは各行の先頭から最初のカンマまでの部分にあたります。この部分だけを取り出すにはstrchr()関数を使用すればよいでしょう。

```
strchr(STR0,DATA)
```

で文字列STR0からDATAにあたる文字コードの文字が最初にみつかる位置を教えてくれます。具体的には，

```
strchr("ABCDEFG",'E')
```

のように使います。この例では4が返ってきますね。ちなみに先頭にあったときは0です。

なお，先ほどのDATAにあたる部分にchar型が要求されていることに気をつけてください。X-BASICの場合，

```
"E"
```

は文字列ですが，

```
'E'
```

は数値（Eの文字コード）となります。

これで位置がわかったら，left$()を使えば最初のブロックを抽出するのは簡単にできることになります。結局，

```
left$(STR0,strchr(STR0,','))
```

ですね。

●比較する

こうして2つの辞書ダンプファイルからそれぞれの先頭の単語の読みを得ることができます。それぞれ，y1，y2としましょう。

文字コード順に並べ換えるにはy1，y2のどちらが大きいかというのを比べなければなりません。

手始めに，

```
str y1="山口県"
str y2="山口市"
```

という2つの文字列が与えられたとしましょう。このうち漢字コード列の大きい文字列を表示せよというプログラムを作れといわれた場合のことを考えます。

＊　　＊　　＊

とりあえず10秒ほど考えてみてください。

＊　　＊　　＊

そろそろ頭の中でだいたいの方針がまとまったと思います。

なお，定数値なので……とばかりに最適化して，

```
10 print"山口市"
20 end
```

表2　漢字コード表(部分)

```
824F:2330:0316: ０ １ ２ ３ ４ ５ ６ ７ ８ ９
825F:2340:0332:    Ａ Ｂ Ｃ Ｄ Ｅ Ｆ Ｇ Ｈ Ｉ Ｊ Ｋ Ｌ Ｍ Ｎ Ｏ
826F:2350:0348: Ｐ Ｑ Ｒ Ｓ Ｔ Ｕ Ｖ Ｗ Ｘ Ｙ Ｚ
8280:2360:0364:    ａ ｂ ｃ ｄ ｅ ｆ ｇ ｈ ｉ ｊ ｋ ｌ ｍ ｎ ｏ
8290:2370:0380: ｐ ｑ ｒ ｓ ｔ ｕ ｖ ｗ ｘ ｙ ｚ
829E:2420:0400:    ぁ あ ぃ い ぅ う ぇ え ぉ お か が き ぎ く
82AE:2430:0416: ぐ け げ こ ご さ ざ し じ す ず せ ぜ そ ぞ た
82BE:2440:0432: だ ち ぢ っ つ づ て で と ど な に ぬ ね の は
82CE:2450:0448: ば ぱ ひ び ぴ ふ ぶ ぷ へ べ ぺ ほ ぼ ぽ ま み
82DE:2460:0464: む め も ゃ や ゅ ゆ ょ よ ら り る れ ろ ゎ わ
82EE:2470:0480: ゐ ゑ を ん
833F:2520:0500:    ァ ア ィ イ ゥ ウ ェ エ ォ オ カ ガ キ ギ ク
834F:2530:0516: グ ケ ゲ コ ゴ サ ザ シ ジ ス ズ セ ゼ ソ ゾ タ
835F:2540:0532: ダ チ ヂ ッ ツ ヅ テ デ ト ド ナ ニ ヌ ネ ノ ハ
836F:2550:0548: バ パ ヒ ビ ピ フ ブ プ ヘ ベ ペ ホ ボ ポ マ ミ
8380:2560:0564: ム メ モ ャ ヤ ュ ユ ョ ヨ ラ リ ル レ ロ ヮ ワ
8390:2570:0580: ヰ ヱ ヲ ン ヴ ヵ ヶ
```

ですませた人，もう少し地道にやってみてください。

さて，これらの文字列は文字コードで示すとそれぞれ，

8E52 8CFB 8CA7
8E52 8CFB 8E73

となります。

だからといって，文字列の長さを調べてforループを回し，1バイトずつ，

```
for i=1 to len(y1)
  if asc(mid$(y1,i,1))＞asc(mid
......
   ：
next
```

などとやっていては一人前のBASICプログラマとはいえません（ま，動けば勝ちという話もありますが）。

文字列を構成する文字コード列の大きさを調べるにはどうするのかというと，

y1>y2

のような感じでいいのです。知ってる人には当たり前ですが，関係演算子は数値演算にしか使えないものだと思い込んでいる人も多いと思います。こういうのは知っているといないとで手間がずいぶん違ってくるものです（特にインタプリタ上では）。

マージプログラム1，2

ということで簡単に考えればリスト1のようなものができあがります。ずいぶん短いリストですが（さらにいえば100～130行はなくても動く），この程度の規模のツールならこんなもんでしょう。このままでも一応動作します。

なにか問題があるのかというと，両方のファイルに同じ単語があったときにその両方ともを入れてしまうのです。辞書メンテを使えば単語の重複は無視されますのでこのままでも使えなくはないのですが，テキストファイルをいじるときのことを考えると，どうにもまずい仕様です。

そこで重複分は片方だけを入れるようにしましょう。

先ほどのものは単語の読みだけをチェックしていましたが，今度は単語全体を見なければなりません。正確にいえば，「学習情報を除いた単語の部分」です。

まず，学習情報の違う同じ単語があった場合の処理を決めなくてはなりません。

より使い込まれたほうを優先するか，単にマージのときの指定順序で決めてしまうかという2通りが考えられます。ここでは，辞書ダンプファイルで先に指定されたものを優先するという仕様にしてあります。経

リスト1

```
 10 int i,j,k,f,g,h
 20 str s1[128],s2[64]
 30 str t1[128],t2[64]
 40 str cr,n1,n2,n3
 50 cr=chr$(13)+chr$(10)
 60 input "dic1,dic2,output filename";n1,n2,n3
 70 f=fopen(n1,"r")
 80 g=fopen(n2,"r")
 90 h=fopen(n3,"c")
100 freads(s1,f)
110 freads(t1,g)
120 s2=left$(s1,strchr(s1,','))
130 t2=left$(t1,strchr(t1,','))
140 repeat
150   if s2<t2 then ss() else tt()
160 until k<1
170 fcloseall()
180 end
190 func ss()
200   fwrites(s1+cr,h)
210   k=freads(s1,f)
220   s2=left$(s1,strchr(s1,','))
230 endfunc
240 func tt()
250   fwrites(t1+cr,h)
260   k=freads(t1,g)
270   t2=left$(t1,strchr(t1,','))
280 endfunc
```

リスト2

```
 10 int i,j,k,f,g,h
 20 str s1[128],s2[64],s3
 30 str t1[128],t2[64],t3
 40 str cr,n1,n2,n3
 50 cr=chr$(13)+chr$(10)
 60 input "dic1,dic2,output filename";n1,n2,n3
 70 f=fopen(n1,"r")
 80 g=fopen(n2,"r")
 90 h=fopen(n3,"c")
100 s2=s0()
110 t2=t0()
120 repeat
130   if s2<t2 then ss()
140   if s2>t2 then tt()
150   if s2=t2 then se()
160 until k<1
170 fcloseall()
180 end
190 func ss()
200   fwrites(s1+cr,h)
210   s2=s0()
220 endfunc
230 func tt()
240   fwrites(t1+cr,h)
250   t2=t0()
260 endfunc
270 func se()
280   s3=left$(s1,len(s1)-strrchr(s1,','))
290   t3=left$(t1,len(t1)-strrchr(t1,','))
300   if s3=t3 then t2=t0()
310   ss()
320 endfunc
330 func str s0()
340   k=freads(s1,f)
350   return(left$(s1,strchr(s1,',')))
360 endfunc
370 func str t0()
380   k=freads(t1,g)
390   return(left$(t1,strchr(t1,',')))
400 endfunc
```

ファイル処理

テキストファイルなどのデータを扱うにはファイル処理をしなければなりません。一般のBASICでのファイル処理というのはややこしい概念を強いられますが，X-BASICでは明快です。ランダムアクセスだのシーケンシャルアクセスだのといった区別もありません。ファイル処理はX-BASICの得意分野といえるでしょう。

ファイル処理の基本は，

開く
いじる
閉じる

の3ステップです。

全体を通して意識する必要があるのは，

ファイル番号
ファイル内のデータポインタ

の2つだけです。

X68000ではファイルはOSが管理しています。OSへのファイル処理開始の宣言としてファイルのオープンという操作が必要になります。ファイルのオープンは処理したいファイル名を与えることで行われますが，そのときOSが返してくるのがファイル番号です。これはたくさんのファイルを開いたときに，どれに対して操作を加えたいかという指定のために使われると思っておいてください。

番号ではなくて直接ファイル名で操作すればいいじゃないかと思う人もいるかもしれませんが，ファイル名指定では同じファイルを複数回開いたときに（普通は危険なのであまりこういうことはしません），混乱が起こることは容易に想像できるでしょう。

こうしてオープンされたファイルにひとつにつき1個ずつデータポインタが割り当てられます。データポインタの中身はファイルの先頭からの位置を表しています。ファイルの読み書きはすべてデータポインタの位置から行われるものだということを覚えておいてください。

ファイル自体はカセットテープのようなもので，タイムカウンタにあたるものがデータポインタです。

で，意識するにせよ，しないにせよ，ファイルの読み書きを行ったときは必ずファイルポインタが変化しています。これを故意に変えることでファイル内の任意の位置がアクセスできるようになるわけです。

SX-WINDOW上でX-BASICを使っていると「ファイルが開いている状態」がひと目でわかります。途中でブレイクして開きっぱなしになった場合は，

fcloseall()

で閉じてください。

験的にこちらのほうが使い勝手がよいと思われるからです（ダンプを繰り返したときの学習情報は一部あてにならないからという説もある）。

strrchr()という関数が用意されています。これはstrchr()と同様の使い方で，DATAが最後に出た位置を返す関数です。

動作をイメージしてみましょう。

安定して動くマスター辞書ファイルDIC0と普段使い込んでいる辞書DIC1があったとします。コンパイルしてきちんとコマンド化し，

DICMERGE　DIC0 DIC1 DIC2

としたとき，DIC2は「DIC1で追加登録された単語を加えたマスター辞書」となります。

使用中に辞書が壊れることはよくあることです。DIC1が壊れて一定のブロックの単語が変換できなくなったとしましょう。この場合は逆に，

DICMERGE　DIC1 DIC0 DIC2

とすることで，DIC2は「現在使用中の辞書の壊れてアクセスできなくなった単語を修復した辞書」となります。

*　　*　　*

辞書は学習すれば賢くなるとは限りません。むしろ馬鹿になることが多々あります。かといって，学習なしで辞書を使うのは非常に不便なものです。このような状況を回避するためにもマスター辞書ファイルを持っておくことは有効でしょう。

マスター辞書ファイルは「少しだけ学習した辞書」を使います。要するに「極端に使いにくくはない安全な辞書」を確保しておくわけです。

学習された単語の抽出

どのような局面でも最適の変換をする辞書を求めるのは無理というものですから，どうしても学習は必要になります。学習すれば，それはそれでまたほかの局面で不都合が起こってきます。

学習というのがなにをやっているのかというと，おそらく単語の切り直しなどが行われたときには内部に新規の単語として登録され，すでに登録されている単語のなかからどれかを選ぶと，同じ読みの単語のなかで学習コードが最大のもの＋1がその語の学習コードとして書き込まれるのでしょう（細かいところは未確認）。

内部登録された単語を抜き出せれば非常によいのですが，ちょっとそこまで調べきれていませんので，学習コードのほうをなんとかしてみましょう。

通常の単語では0～999までの学習コードがあり，これらは基本的に増える一方ですから，やがて学習内容に大きなばらつきが出てきます。使い込んでいくとその使用状況では賢くはなりますが，別の状況ではとんでもない単語を出してくる可能性も増えるわけです。

ということでなんらかの学習が行われた単語を抽出するプログラムを作ってみることにします。

リスト3です。

すでにこれまでの解説でほぼ理解できるでしょう。

ダンプされた辞書を見ると，品詞コードは2桁の品詞情報と3桁の学習情報といった数値を表す文字列で構成されています。学習がされていない場合は学習情報はなし，という具合です。そこで品詞コード分を取り出して，頭の2文字以外を取り出せば学習情報だけが得られます。

right$()などの組み合わせで文字列のなかから数値に当たる部分を取り出します。これだけだとまだ単なる文字列です。

"025"

と，

025（実際には25）

は違いますので型変換という操作が必要です。

数値を表しているはずの文字列を数値に変換するのがval()関数です。

val()関数を使うと，文字列がnull（要するに空っぽ）のときにエラーが発生しますので，ここでは頭に"0"を加えてこれを回避しています。

もちろんif文で場合分けしてもかまいませんが（見た目にはそのほうがわかりやすいかも），個人的な趣味でif文を使わなくてすむところではなるべく使わないようにしているだけです（構造上重要な分岐には使う）。

こうしてプログラムは完成しました。

ここでは動作に必要な部分だけを解説しましたが，ものがツールですからエラーチェックもしておいたほうがいいのはいうまでもありません（辞書メンテはダイアログ内に表示されるのでチェックが困難）。

本来なら項目数が4つあるか，読みがソートされているか，などをチェックすべきでしょうが，今回は割愛してあります。皆さんで考えてみてください。

また，コンパイルを前提にするならコマンドラインパラメータの処理も加えたほうがよいでしょう。

*　　*　　*

なんかBASICの話というよりも辞書の解説みたいになってしまいましたが，必要なツールはたいていX-BASICで作れるということはわかっていただけたのではないでしょうか（おお，強引だ）。

ツールに要求したいものをよく煮詰めておくことが大切です。といっても最初から効率のよいものが作れるわけもないですから，やはり試行錯誤ということですか。

リスト3

```
 10 int i,j,k,f,g,h
 20 str s1[128],s2[64]
 30 str cr,n1,n3
 40 cr=chr$(13)+chr$(10)
 50 input "dic,output filename";n1,n3
 60 f=fopen(n1,"r")
 70 h=fopen(n3,"c")
 80 repeat
 90   k=freads(s1,f)
100   s2="0"+right$(s1,len(s1)-strrchr(s1,',')-3)
110   if val(s2)>0 then fwrites(s1+cr,h)
120 until k<1
130 fcloseall()
140 end
```

データは2種類……

今回のマージプログラムで扱う文字列は2つ。基本的な部分ではそれぞれ対等なものですから，文字列y1とy2で同じような操作が必要になります。こういうときには関数を使って同じような処理をまとめるのが筋ですが，リスト1では同等のものを2つ並べてみました。

こういう作り方をしていると，後々の拡張でその部分と周辺まで作り直さなければならなくなることも往々にしてあるのですが，プログラム作成初期段階で動作を確認するまでは，できるだけ単純な構造のプログラムを作ることを重視してもかまわないでしょう。

いくばくかでも汎用性を持った部品を作ろうとすることは悪いことではありません。そういう癖をつけておけばより拡張性が高いプログラムを作ることができると思います。

仕様をケチったところで最適化しながらプログラムを作っていると，あともうひと押しのところで拡張にいきづまってしまうことが多々あります（反省）。

そういうハマリ状態になったときには，大改造などはあきらめて最初から作り直すことでよりすっきりしたプログラムを作ることができます。「動くもの」に少しずつ手を加えていくのが開発の基本ともいえますが，そういう状態になったときはプログラムはかなりツギハギ状態になっているものですから。

カードゲームの基本処理

これから始めるSX-BASIC

Nakano Shuichi 中野 修一

少しずつ環境整備の進んでいるSX-BASIC
そろそろ手を出してみたくなった人も多いのではないでしょうか
ここではカードの扱いを中心にまとめてみました

少しずつSX-BASICがその本領を見せ始めてきました。SX-BASICの意義は，単に「SX-WINDOW上でBASICが動く」ということよりも，それを核として統合されたマルチタスク/マルチウィンドウ環境というものが構築できるという点にあります。対応ソフトが増えてくるにしたがって，アプリの1つひとつが作る点を線に結び，相互にネットワークを結ぶことができるのです。1，2年前のSX-WINDOW特集で観念的なお話ばかりしていた世界がようやく形を成し始めたといえるでしょう。

しかしもちろん，タスク制御以外にもSX-BASICの利点はあります。簡単にSX-WINDOWアプリを作成できることです。X-BASICがある程度使える人ならばSX-BASICのプログラムは簡単に作成できます。文法がほとんど同じですから当然といえば当然の話ですね。

プログラミングの手順

SX-BASICのプログラムを作成する手順は，だいたい，

ウィンドウデザイン
アイテムの設定
アイテムのイベント処理
プログラム各部の作成

のようになります。

では順を追って見てみましょう。

●ウィンドウデザイン

まずウィンドウデザイナでだいたいの感じをつかみます。同時にパターンエディタを立ち上げて，ビットマップパターンでボタンなどをデザインしてもいいでしょう。最初から気合入れて作り込んでもかまいませんし，とりあえず必要そうなものを適当に並べたものでもかまいません。まずは画面から作ります。

●アイテムの設定

ウィンドウ上に置かれたテキストやボタン類の基本設定を行います。各アイテムを選択してメニューから“property”を選択してください。設定ウィンドウが現れるので適当に設定してください。ここをいろいろ変えてみれば，どんなことができるのかというのも少しはわかるのではないでしょうか。

●アイテムのイベント処理

次に各種アイテムの受け取るイベントに対しての処理ルーチンを作ります。まあ，とりあえず全部のアイテムをダブルクリックして回ってください。

ここまですんだらタイトルバーのところのメニューから“save file”を選び，データをセーブしておいてください。

●プログラム各部の作成

ウィンドウデザイナはとりあえず置いといて，シャーペン.Xを立ち上げて先ほどセーブしたファイルを読み込みます。ウィンドウデザイナ上でもプログラミングはできるのですが，操作性などを考えるとプログラムの記述自体はシャーペン上で行ったほうがよいでしょう。

読み込まれたプログラムを見ると，

▼

のついたアイテム設定の部分と，各種アイテムのイベント処理関数が並んでいるのがわかるはずです。

イベント処理部分は，アイテム名に，

.click

のようなイベントの種類が並んでいます。SX-WINDOWのプログラムはイベントに対して反応する形式になりますから，あとはそれぞれの処理に対応させたいプログラムを作っていけばよいことになります。

カードを使ってみる

なにかサンプルを作ってみましょう。

適当な大きさのウィンドウ内に適当に2つのビットマップを配置してください。シャーペンでちょこちょこといじって，

```
▼Window Size (600,300),0,1,46,
TEST
openres("card")
Bitmap1.id=1001
Bitmap2.id=1025
▼8, Bitmap 1 ( 30 4,14,392,208),
0,0,0,0,
func Bitmap1_Click()
endfunc
▼8, Bitmap 2 ( 432,22,528,212),
0,0,0,0,
func Bitmap2_Click()
endfunc
```

のようにします(座標値などは適当)。あと，どこかのドライブにもみじ狩りPRO-68K収録のcard.lbを置いといてくださいね。

SX-BASICを立ち上げてこのファイルのアイコンを放り込みます。面倒な人はアイコンメンテで*.SXBのアイコンに，

実行ファイル　SXBASIC.X
実行オプション　-F%

を登録しておくとよいでしょう。

ファイルが実行されてカードが2枚出てきたはずです。

ここでの中心は，

openres("card")

の処理です。これはカードデータを収めたリソースをアクセスするためのものです。もともとSX-BASICでは，基本ウィンドウ作成時にプログラム名と同じ系列の名前のリソースファイルが呼び出されます。そのプログラム独自のリソースはそこで読み込まれます。

みんなが持っていたのでは効率の悪いものは共通リソースとしてまとめることができます。こういったカードデータ以外にも，ピクチャーボタンやアクセサリ的なビットマップなどが揃うとウィンドウ環境独特の味気なさも薄めることができるでしょう。

ただ，あまり頻繁にリソースを切り換え

るのも気持ち悪いので，openres()関数はプログラムの先頭あたりで一度だけ使うもの，くらいに思っておいてください。

さて，次の処理です。シャーペンで，Bitmap2_Click()関数の中身として，

```
print "テスト"
```

を加えてください。OPT.1+Sでセーブし，SX-BASICのウィンドウをアクティブにして，OPT.1+Rとすると新しいファイルを読み込み実行します。以下，このような手順でプログラムをエディット/実行していきます。

カードをつかんで動かすようにしてみましょう。

SX-BASICが最初に発表されたときのサンプルでビットマップのドラッグをやってみたものがありましたが，あのように正確にカードの位置を保存するには定義時の位置を記録したグローバル変数が必要です。

そういうのはちょっと面倒なので，ここではカードをつかんでドラッグ動作に入った瞬間にマウスカーソルがカードの中央になるようにカードを移動させることにしましょう。

カードの大きさは48×64（ただし縦位置のもの）ですから，カードの表示座標は，

```
(mousex-24,mousey-32)
```

となります。以下，マウスの左ボタンが離されるまで処理を続けます。

```
func Bitmap1_Click()
int x,y
x=mousex-24:y=mousey-32
repeat
if not((x=mousex-24) and (y=mousey-32)) then {
  x=mousex-24:y=mousey-32
  Bitmap1.move=x,y,0,0
}
until mousel=0
endfunc
```

のような感じになります。

実際やってみるとこれは結構重い処理だということはわかると思います。途中にdi()とei()を入れることでかなり軽快にはなりますが，その代わり処理中はほかのタスクをすべて止めることになります。

*　　*　　*

だいたいおわかりのように，イベントに対応したサブルーチンを揃えていくことがSX-BASICプログラミングの基本になります。ウィンドウアイテムに対する操作や処理などはすべてウィンドウエンジンが吸収してくれていますので，表示や入力などの基本ユーザーインタフェイスに気を遣うことなくアプリケーションの作成ができます。

ダブルクリック

SX-BASICは普通のクリックはイベントで返してくれるのですが，ダブルクリックが取りにくいシステムになっています。

ダブルクリックはウィンドウプログラムでは多用される（であろう）処理なので，ひとつそれっぽいものを作ってみましょう。

先ほどのプログラムの先頭（▼Window～の直後）にグローバル変数として，

```
str tim
```

を定義してください。次に，先ほどの関数の先頭に，

```
if tim=time$ then double()
tim=time$
```

を加えます。あとは，どこかに，

```
func double()
? "ダブルクリック"
beep
endfunc
```

という関数を作っておいてください。

ここで採用しているのは1秒間に2回クリックが発生したらダブルクリックとみなすという強引な処理です。しかもX-BASICと同じ1秒ごとの時計を見ています。よって途中で秒が変わると取りこぼしが発生してしいます。「これではあまりにも……」と思う人は以前石上氏がやっていたようにIOCSコールで1/100秒タイマを読んでくるのがよいでしょう。

アイテムの配列

ここでBitmap2のほうも同じ処理にしてみましょう。

同じようなものをまた作るのはパスして，アイテムを配列化してみます。ウィンドウデザイナで配列化するのも面倒なので，テキストをいじって直接指定してみましょう。要は同じアイテム名にしてindexを設定するだけです。Bitmap1の参照部分にはすべてindexの指定を付け加えます。

Bitmap2の指定部分だったところは，

```
▼ 8, Bitmap 2 ( 432,22,528,212 ),
0,0,0,0,
```

から，

```
▼ 8, Bitmap 1 ( 432,52,528,212 ),
1,2,0,0,
```

に変わります。

アクセスするには，

```
Bitmap1 [2] .move= ～
```

のように［index］を指定してください。

最後に

TSEnd()という関数はマニュアルには載っていたのに実際には装備されていなかったので大変混乱を招きましたが，最新版ではようやくサポートされました。ただし，名前が変更になり，

```
endtask(N)
```

となりました。タスク番号Nのタスクを終了させます。

さて，SX-BASICを使っていて気づくのは，できないものも多い代わりに，なにも考えずにできるものも多いということです

X-BASICならばコンパイルを前提に考えますので，少々複雑なことでもX-BASICで記述してしまえます。SX-BASICではそうはいきません。その分，SX-BASICではSX-WINDOWシステム自体やウィンドウエンジンがより高度な処理を提供してくれています。

結果として，SX-BASICではサブプログラムとの連係が重要であり，容易に行えるということです（もちろん，対応したサブプログラムがあれば，ですが)。まず，ウィンドウエンジンがSXアプリのスケルトンを代用してくれますし，ゲームキットのようにもっと高度なことをやってくれるツールを増やせば，SX-BASICでできることもどんどん広がります。X-BASICでは外部関数で強化されるものはあくまで「道具」の位置にありましたが，SX-BASICではそれ自体が意志を持った「専任スタッフ」を傘下に抱えているような感じです。

基本的に，X-BASICは処理を記述するものですが，SX-BASICは処理をつなぐものでした。高度なことが簡単にできる代わりに，エンジンの仕様外の事項にはまったく手だしできませんし，それだけのウィンドウエンジンや連係するプログラムについての理解が不可欠になってきています。

しかし，もみじ狩り版では，さらにコードリソースによる拡張にも対応し，X-BASICの外部関数のような感覚でSX-BASICの機能を拡張できるようになりました。本格的な「～エンジン」という規模でなくても機能拡張ができるわけです。タスク間通信でのオーバーヘッドがなくなりますから，こちらのほうでもいろいろ面白いことができるかもしれません。

さらに周辺アプリとの連係が加われば SX-BASICの世界はいっそう広がっていくことでしょう。

テキストグラフィックを扱う

TGRAPH.FNC

Asakura Yuji 朝倉 祐二

これまでずいぶんともったいない使い方をしていたテキスト画面
10月号で紹介しそこねたTGRAPH.FNCを改めて紹介します
これはテキスト画面を活用するために作成された外部関数です

テキスト画面の活用

X-BASICはマウスカーソルを表示する，しないにかかわらず，OSであるHuman68kの機能をそのまま引き継いで，512KバイトあるテキストVRAMの半分の256Kバイトをマウスカーソル表示専用に確保しています。

そのためテキスト表示は4色とかなり貧弱なものになっていますが，512Kバイトをフルに使い切ればテキスト画面で16色表示が可能となるはずなのです。

10月号の付録ディスクに収録したTGRAPH.FNCはX-BASICでまるでサポートされていなかったテキストグラフィックを扱う外部関数です。

コマンドの説明

これからTGRAPH.FNCに含まれる関数群の説明をします。最初に各関数に与えるパラメータで共通仕様のものについて，指定可能な数値の範囲を挙げますのでこれを踏まえて各解説を読んでください。

●座標の指定　x1，y1，x2，y2（int型）

0～1023で指定。範囲外の数値を指定するとエラーになります。

●描画色の指定　col（char型）

0～15で指定。範囲外の数値を指定するとエラーになります。

＊　＊　＊

・tcls()

テキスト画面をクリアします。

・tline(x1,y1,x2,y2,col)

(x1,y1)－(x2,y2)を結ぶ線分をカラーコードcolで描画します。

・tfill(x1,y1,x2,y2,col)

(x1,y1)－(x2,y2)で示される領域内をカラーコードcolで表示します。

・tpset(x1,y1,col)

(x1,y1)にカラーコードcolのドットを表示します。

・tpoint(x1,y1)

(x1,y1)に表示されているカラーコードをchar型で返します。

・tput(x1,y1,dx,dy,ary1,［mode］)

(x1,y1)にあらかじめchar型1次元配列に格納してあるパターンサイズdx×dyのパターンを表示します。dx，dyはパラメータチェックをしていませんので，int型で表せる範囲内で指定してください。

char型1次元配列ary1には表示するパターンを格納します。配列の大きさは最低パターンサイズdx×dy−1以上必要です。パターンはドットごとに16色を指定可能で，スプライトパターンの定義と同じ感覚でできますのでわからないことはないと思います。

modeはstr型の引数で，"set"または"xor"が指定できます。"pset"を指定した場合は表示領域にパターンを上書きします。"xor"を指定した場合は，表示領域とパターンの各ドットごとにxorをとった結果を表示します。

老婆心ながら"xor"でパターンを表示する場合は，同じパターンを2回表示することで，パターンを表示する前の状態に戻すことができます。なお，modeは省略可能で，この場合は"pset"モードで表示します。

ここでバグレポートです。原稿を書いていて気づいたのですけど，dx，dyにマイナス値を指定してもエラーにならないんです。間違ってもマイナス値を指定しないでください。申し訳ない。

・tpalet(col,pal)

カラーコードcolのパレットコードをpalに変更します。palはint型で0～65535の範囲内で指定しないとエラーになります。

・thome(x1,y1)

テキスト画面の実画面における表示画面の左上座標を(x1,y1)にします。x1に大きな座標を指定すると表示が乱れることがありますが，これは仕様であってバグではありません。悪しからず。

XCで使うために

TGRAPH.FNCに含まれる関数を使用したX-BASICプログラムのコンパイル方法を説明します。

まず，付録ディスクに収録されているtgraph.defをBC.Xと同じディレクトリにコピーします。続いて同じく付録ディスクに収められているtgraph.hを，環境変数includeに設定しているディレクトリ，もしくはコンパイルするX-BASICプログラムと同じディレクトリにコピーしておきます。

これで前準備は完了です。

あとは普段X-BASICをコンパイルするのと同じ要領で作業します。最後にリンクするところで，付録ディスク内のtgraph.aをライブラリに追加指定するのを忘れないでください。

あとがき

最後まで黙っていましたが，発表したTGRAPH.FNCに，ペイント，サークルなどグラフィックを扱う代表的な関数が含まれていないことに気づいた読者はたくさんいることでしょう。すいません，私の力不足で締め切りに間に合わすことができませんでした（反省）。編集部の皆様や読者諸氏には大変申し訳ないことをしました。これらの関数は作成を諦めたわけではありません。今月外部関数とライブラリの作り方のところで作成したテキストの高速PUT関数などもその一環をなすものです。次の機会に発表すべく開発していますので，いましばらくお待ちください。

ハイレベルなピコピコゲームを

XSPRITE.FNC使用ガイド

Itou Masahiko 伊藤 雅彦

**スプライトの潜在能力を引き出すXSPRITE.FNC
今回はマニュアルとして使用上の注意点として紹介しましょう
非常に高度な処理を簡単に使いこなすことができます**

先月号の付録ディスクに入っていたXSPRITE.FNC。この外部関数はX-BASICによるお気軽ゲームプログラミングをお手伝いするものとして紹介させてもらったわけですが，どんなもんでしょう。先月は概略しか説明していなかったので，今回は使い方をもうちょっと丁寧に説明しましょう。できれば先月号の関数リファレンスも参照しながら読んでくださいね。

初期化など

XSPRITE.FNCの関数は，スプライトのスライド移動を行う関数ですが，スライドを開始する関数をいきなり実行してもエラーが出てしまいます。まず最初に初期化関数sp_xinit()を実行しなくてはいけないのです。この関数は内部ワークのクリアなどをするものです。初期化せずにほかの関数を実行しようとしても未初期化エラーが発生するようになっています。

また，スライド移動は垂直同期割り込みを使ってスプライトの表示位置を変えることによって実現されていますから，スライドを行う前にはsp_slon()を実行して割り込みを開始させます。

つまり，スプライトをスライドさせるには，sp_xinit()とsp_slon()を実行しないことには始まらないということです。さらに，sp_xinit()を実行したあとに画面モードを変えたり，標準のスプライト初期化関数sp_init()を実行したりするとまずいので，画面関係の初期化処理は次のようにしてください。

```
screen ×，×，×，×
sp_init()
sp_xinit()
(スライド開始前の処理)
sp_slon(×)
```

ここで，sp_slon()の引数の値ですが，普通は0にします。0にすると，垂直帰線期間に入るたびに割り込みがかかります。高解像度(31kHz)画面のときは1秒に約55回，標準解像度(15kHz)画面のときは1秒に約60回の割り込みがかかることになります。引数を1にすると，割り込みが垂直帰線期間2回に1回だけになって，以下，引数の値を1大きくするごとに3，4，5……回に1回だけ割り込みがかかるようになります。

割り込み周期は短くしたほうがスプライトが滑らかにスライドしますから，引数は0がいいというわけです。ただし，どうも割り込み処理にCPUパワーを取られていけないという場合は，引数を1にしてみるのもいいかもしれません。一定時間あたりの割り込み回数が半分になりますから，割り込み処理の負担は半分になります。割り込み処理が重過ぎて困るっていうケースが実際にあるのかどうかはわかりませんけど。

それから，プログラム終了時にはsp_sloff()関数で割り込みを停止しましょう。割り込みがかかりっぱなしになっていると，CPUパワーが無駄に使われてしまいます。

スライド関数

スプライトをスライドさせる関数は2種類あります。sp_slidep()とsp_slidev()です。

sp_slidep()から説明しましょう。これはスプライトを指定した座標にスライドさせるものです。

```
sp_slidep(0, 150, 120, 32)
```

を実行すると，プレーン番号0のスプライトが現在の表示位置から座標(150，120)にスピード32でスライドします。

ここでの，スピードの値なんですが，できるだけ32の倍数を指定してください。というのは，そうするとスプライトがスムーズにスライドするからです。スピードの値は32が基準になってまして，スピード32でスライドさせると，スプライトは割り込み周期ごとに1ドットずつ動いていきます。ここでスピードを33にすると，割り込み32回中31回は1ドット動きますが，1回は2ドット動くことになります。すると，これは実際に試していただけるとわかるんですが，2ドット動くときだけなんだかカクッ，カクッとスライドがガタつくのです。このガタつきを防ぐには毎回同じドット数だけ動くようにすればいいわけで，そのためにはスピードの値を32の倍数にしてあげればいいのです。

それから，斜め方向にスライドする場合，スピードの値は，x，y方向のうち移動量が大きいほうの方向へのスライドスピードになります。たとえば，図1のようにスピード32で各方向にスライドさせた場合，1)と2)の例ではx方向のスピードが32になり，4)と5)の例ではy方向のスピードが32になります。3)の例ではx，y方向の移動量が同じですから，両方とも32のスピードになります。

一方，もうひとつのスライド関数sp_slidev()では，x，y方向の移動スピードを直接指定してスライドを行うことができます。たとえば，

```
sp_slidev(0, 32, 64, 50)
```

で，x方向32，y方向64のスピードで右下にスライドします。最後のパラメータはスライドする時間を指定するもので，実際のスライド時間は50×(割り込み周期)になります。が，別の表現をすると，この例では割り込み処理が入るたびに右に1ドット，下に2ドット動くんですが，この移動を50回やるということなんです。つまり，スプライトが右下にスス一ッとスライドして，右に50ドット，下に100ドットのところまできたところでピタッと止まるというわけです。

2つのスライド関数の違い，わかっていただけましたか？　大ざっぱにいうと，sp

_slidep()は「あの場所にスライドしろ」というもので，sp_slidev()は「あっちの方向にスライドしろ」という感じになります。うまく使い分けてくださいね。ちなみに，サンプルゲーム”XSP”では，敵機や自弾，煙を動かすのにはsp_slidep()を使って，敵弾の移動にはsp_slidev()を使っています。敵機の動きなんか，いかにもsp_slidep()を使ってますって感じでしょう。敵弾は自機の位置に応じて発射する方向を決めています。つまり方向が問題になっているんですから，sp_slidev()のほうが便利です。

それから，ついでにsp_loc()関数も説明しておきましょう。これはスライド移動ではなく，普通の(瞬間？)移動をする関数です。これは標準の（というかSPRITE.FNCの）sp_move()やsp_set()でもできることですが，XSPRITE.FNCの関数を使うときにはこれらの標準関数を使って移動をしてはいけません。ただし，パターンコードやパレットブロックなどを変えるのに使うのはOKです。あくまでも，移動するならsp_loc()で頼んます，ということです。

また，sp_loc()は位置指定を省略して，

sp_loc(0)

などとして，スライド中のスプライトを止めるときにも使います。

ジョイスティック操作

サンプルゲームでは，敵機をはじめ，ほとんどのスプライトを上記のスライド関数で動かしていますが，自機だけは例外です。自機を動かすのには，sp_stkon()という便利な関数があるんです。これは，いわばマイキャラ移動専用のスライド関数で，スプライトをジョイスティック操作に応じてスライドさせます。たとえば，

sp_stkon(1, 0, 2, 3)

を実行すると，ジョイスティック1のスティック操作に応じて，プレーン番号0のスプライトが上下左右に（もちろん斜めにも）動くようになります。3，4番目の引数は動くときのx，y各方向の移動ドット数です。スプライトが表示できる画面モードではドットが横長になっていますから，この例のようにx方向へは2ドットずつ，y方向へは3ドットずつ動くようにすれば，見た目では縦横とも大体同じスピードで動いてくれます。

で，ここで問題になるのがスプライトの動ける範囲です。スティック操作の通りに動くからって，どこまでも，画面の外にまでも動いていっちゃったら話になんないでしょう。

この関数では，移動可能範囲をグラフィック画面で設定します。まず，あらかじめグラフィック画面を実画面512×512のモードにしておきます。色数は16，256，65536色どれでも構いません。screen命令で，

screen 0, 1, 1, 1

などとやります。第2パラメータで512×512モード，第4パラメータでグラフィック関数使用可を指定すればOKです。そうしておいて，たとえば移動可能範囲を(16, 16)－(255, 255)の矩形領域にしたい場合は，

fill(16, 16, 255, 255, 1)

を実行します。これで移動可能範囲の設定は完了です。あとは，必要な初期設定をしてから，

sp_loc(0, 136, 136) /* 中央に移動
sp_stkon(1, 0, 1, 1) /* 操作開始

とでもしてやれば，範囲内をスプライトが動き回ります。

もうちょっと説明らしくいうと，グラフィック画面のスクリーン0に，移動可能にする範囲にだけ奇数のパレットコードを書き込めば，そこが移動可能範囲になる，ということなんです。screen命令を第4パラメータ＝1で実行すると，グラフィック画面がクリアされますから，パレットコードは画面中すべて0になります。そこにパレットコード1で塗り潰し矩形を描けば，そこが移動可能範囲になるわけです。

もちろん，移動可能範囲は矩形に限りません。三角形の範囲だって，丸っこい範囲だって，グラフィック画面に好きなように描けば望み通りです。さらに，縦線，横線を描けば，スペースインベーダーの砲台のような動きや，迷路型の動きも楽勝です（図2）。

グラフィック画面を移動範囲指定に使うことで，グラフィックが1面使えなくなるという欠点は確かにあります。でも，256色モードならもう1面あるからいいでしょう。それに，たいていは表示画面を256×256にするでしょうから，範囲指定に使うスクリーン0だって左上1/4しか使わないわけで，表示範囲をずらせばスクリーン0もグラフィック表示用に使えます。移動可能範囲を自由に指定できることから，いろんなタイプのゲームに対応できるというメリットのほうがずっと大きいでしょう。

ところで，このマイキャラ移動関数があるからといって，マイキャラ移動にこの関数を必ず使わなくちゃいけないということはありません。奇抜な操作系に味があるというゲームもありますので，マイキャラ移動にsp_stkon()を使うことばかり考えないようにしましょう。

スプライト同士の連結

スプライトの大きさは16×16ドット。当然，このサイズより大きいキャラクタを使いたいときだってありますから，そんなときは複数のスプライトを使って表示しなくてはいけません。サンプルゲームでも，自機と敵機はスプライト2つを横に並べて表示しています。

そういう場合，キャラクタをスライドさせるときにキャラクタを構成する全部のスプライトについてスライド関数を実行しな

図1 スピードの設定値と実際の移動スピード

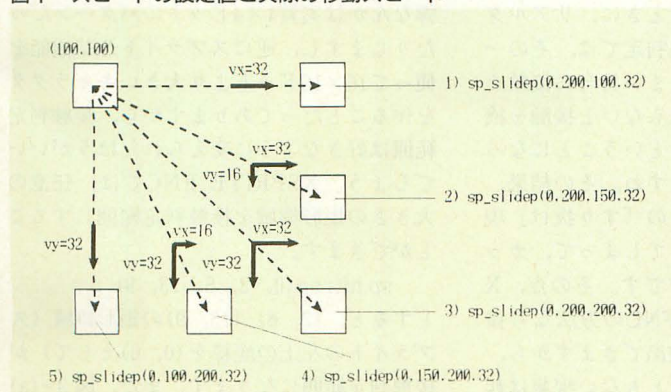
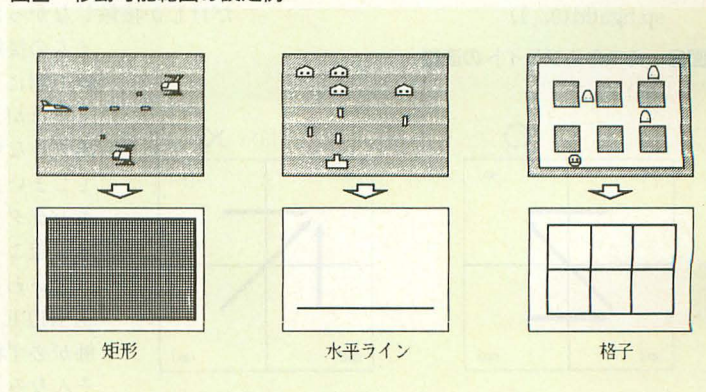



図2 移動可能範囲の設定例

ければいけないとしたら，実に不便でしょう。そんなわけで，スプライト同士を連結してひとつのスプライトのようにスライドさせるための関数，sp_hang()があります。

sp_hang(0, 1, 16, 0)

これで，スプライト0の右16ドット，下0ドットの位置，つまり右隣りにスプライト1がくっつきます。くっつくといっても，正確にはこの関数を実行しただけでは，実際にスプライト1がスプライト0の右隣りには移動しません。この後，スプライト0に対してsp_slidep()，sp_slidev()，sp_loc()，sp_stkon()のどれかを実行したときに，実際にくっつきます（sp_stkon()の場合は，スティックを倒してスプライトが動いたとき）。ま，これはそれほど気にする必要はないでしょう。

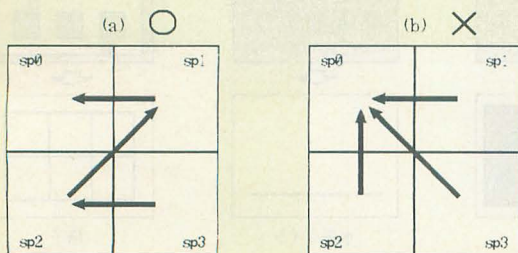
当然，3つ以上のスプライトを連結することもできます。

sp_hang(0, 1, 16, 0)
sp_hang(1, 2, -16, 16)
sp_hang(2, 3, 16, 0)

とすれば，図3-(a)のように連結します。このようにいもづる式につなげるのがポイントで，図3-(b)のようなつなげ方はできません。こうやってスプライトをどんどん連結していけば，どんなにでかいキャラクタだって作れます。あ，スプライト個数の制限はありますけど。

接触判定関数

接触判定機能。ゲーム作りの必需品です。もちろん，ゲーム作成をお手伝いするXSPRITE.FNCには接触判定用の関数があります。でも使い方がちょっと面倒なんで，例を挙げて説明しましょう。

スプライト1～3が敵キャラ，スプライト4がマイキャラで，マイキャラが敵キャラと接触しているかどうかを判定したいとしましょう。まず敵キャラのスプライトを接触判定グループに登録します。

sp_hgadd(0, 1)
sp_hgadd(0, 2)
sp_hgadd(0, 3)

これでスプライト1～3が接触判定グループ0に登録されます。そして，

sp_hiton(4, 0)

で，スプライト4が接触判定グループ0に登録されているスプライトと接触しているかどうか，つまりマイキャラと敵キャラとの接触の判定が始まります。この判定は割り込み処理中に行われます。マイキャラが敵キャラのどれかと接触するまで，割り込みがかかるたびに接触判定をするのです。

で，接触したかどうかをチェックするには，sp_hit()関数の値を見ます。まだ接触がなければ戻り値は255で，接触があったら接触したスプライトのプレーン番号が戻り値になります。

if sp_hit(0)<255 then ～

というような使い方になるでしょう。ただし，接触があったときの処理で接触スプライトのプレーン番号が必要なときは，

hs=sp_hit(0)
if hs<255 then ～

というように，戻り値を変数に保存してください。sp_hit()の戻り値をもう一度見ても，値は255に戻っていますので。

そして，いったん接触があると，もう接触判定は行われなくなりますから，また接触判定をしたいときにはsp_hitoff()をもう一度実行します。

わかったでしょうか。ここで気づいてほしいのは，sp_hit()がリアルタイムの接触状況を返す関数ではないということです。この関数は，sp_hiton()が実行されてから接触があったら，その後また接触状態でなくなっていても，一度接触したスプライトのプレーン番号を返すわけです。

こうなっているのは，リアルタイムの判定では接触を見逃す可能性があるからです。BASICは実行速度が遅いですから，接触チェックの間隔がどうしても長くなってしまいます。すると，スプライトが一瞬の間だけしか接触しなかったときに，リアルタイムの接触判定では，その一瞬の間にうまい具合に接触チェックが入らないと接触を検出できないということになってしまいますね。その結果，キャラクタの「すり抜け」現象が起こってしまって，カッコ悪いわけです。その点，XSPRITE.FNCの方法なら接触が必ず検出できますから，そんなみっともない現象は起きません。

それから，もうひとつ注意してほしいのは，sp_hiton()を実行した直後には接触が検出されないことです。たとえば，

sp_loc(0, 100, 100)
sp_loc(1, 100, 100)/＊sp0, 1が接触
sp_hgadd(0, 1)
sp_hiton(0, 0)　　/＊接触判定開始
print sp_hit(0)　　/＊接触状況表示

とすると，"1"と表示されそうですが，ほとんどの場合は"255"（接触なし）と表示されてしまいます。どういうわけかというと，接触判定は割り込み処理で行われるので，sp_hiton()が実行されてから一度は割り込みがかからないと，接触していても検出されないんです。

これがちと情けない仕様だというのは私も認めます。でも逃げ道はあるので許してください。sp_hiton()実行後に割り込みがかかるまでは接触判定がないということなんですから，割り込みがかかるまで待てばいいわけです。ここでsp_intcnt()関数が役に立ちます。この関数は初期化関数sp_xinit()を実行してから割り込みがかかった回数を返すものです。つまり，割り込みがかかるたびにsp_intcnt()の値はカウントアップされます。だから，

i=sp_intcnt()
repeat: until sp_intcnt()>i

というウェイト処理をsp_hiton()の後ろに入れれば，正しい接触状況を知ることができるようになります。

また，同じ事情で，sp_loc()で移動したスプライトの接触状況も移動直後には検出されません。これもウェイト処理を入れれば逃げられますが，ウェイト処理というのは時間を無駄に食い潰すことなので，本当に必要なところだけに入れてください。たいていの場合はいらないと思います。

ところで，スプライトは16×16ドットですから，接触判定の範囲も初期状態では16×16です。でも，シューティングゲームの弾なんかは実質4×4ドットのパターンだったりしますし，逆にスプライト連結機能を使って16×16ドットより大きいキャラクタを作ることだってありますから，接触判定範囲は好きなように変えられたほうがいいでしょう。XSPRITE.FNCでは，任意の大きさの矩形領域を接触判定範囲にすることができます。

sp_hitrng(0, 2, 6, 13, 9)

とすると，(2, 6)-(13, 9)の矩形領域（スプライトの左上の座標を(0, 0)として）が接触判定範囲になります。また，図3-(a)

図3　4つのスプライトの連結

のように大きなキャラクタを作ったときは,

sp_hitrng(0, 0, 0, 31, 31)

というように，スプライト0の接触判定範囲を(0, 0)−(31, 31)にすれば，スプライト0の接触を調べるだけでキャラクタ全体の接触を調べられるので便利でしょう。

バックグラウンド関数

スプライトをスライドさせたついでに，バックグラウンドもスライドできるようにしてみました。サンプルゲームでやっているような，スムーズな背景スクロールが簡単にできます。

bg_slide(0, 32, 64, 0)

とすると，左へスピード32，上へスピード64でスライドします。引数の指定方法はsp_slidev()と大体同じになっています。違うのは，スピードにプラスの値を指定したときに，sp_slidev()では右下へスライドしますが，bg_slide()では左上へスライドすることです。また，bg_slide()ではスライド時間に0を指定すると，BGがずっとスライドし続けるようになっています。

さらに，このBGスクロールに関しては，ちょいと気張った機能があります。BGの大きさは64×64チップですが，もっと大きな背景を用意してスクロールさせたいこともあるでしょう。64×64といったら，表示画面の縦横各2倍しかないわけですから。そこで，縦横をそれぞれ33〜32767チップの任意の大きさにした仮想BG面を用意して，表示・スクロールできる機能をつけました。

仮想BG面を用意するにはbg_map()関数を使うんですが，その前に配列に仮想BGの画面内容を設定しておく必要があります。仮想BGを200×100チップにしたいときは，

dim char mapdata(200*100-1, 1)

といったように，要素数200×100×2個のchar型2次元配列を宣言します。必ずchar型2次元の配列でないといけないので注意してください。そして，仮想BGの座標(x, y)のチップのパターンデータを，mapdata(y*100+x, 0)とmapdata(y*100+x, 1)に代入します。パターンデータは16ビットですから，上位8ビット（反転指定とパレットブロック）をmapdata(y*100+x, 0)に，下位8ビット（パターンコード）をmapdata(y*100+x, 1)に代入してください。

こうしてすべての座標のパターンデータを配列に設定したら，準備は完了です。ここで，

bg_map(0, 200, 100, mapdata)

とすると，BG0を仮想BGを表示するモード（マップモードと呼ぶことにします）にして，サイズが200×100チップ，配列mapdataに設定した内容の仮想BGが用意されます。このあと，通常のモードのときと同じようにbg_slide()を実行すれば，仮想BG画面がスクロールします。

さて，ここでマップモードの欠点を白状しましょう。実に不便なことに，仮想BG面は内容を書き換えることができません。bg_map()で配列を使って画面内容を設定したら，あとはずっと変えられないのです。一部分でも変えようと思ったら，もう一度bg_map()で仮想BGを設定し直すことになります。標準のbg_put()に相当するような関数があるとよかったなあと私も思いますが，作らなかったのは，もちろん，私の怠慢です。ごめんなさい。

謝り終わったところで，説明を補足させていただきますと，マップモードを解除するには，

bg_map(0, 0, 0, mapdata)

というように，2, 3番目の引数を0にしてbg_map()を実行します。また，マップモード時に仮想BGの表示位置を設定する関数として，bg_mpscrl()が用意してあります。ま，いってみれば標準のbg_scroll()関数のような機能のものです。

その他の関数

XSPRITE.FNCには，スプライトには関係ないけどゲーム作りに役立ちそうだなという関数も加えてあります。

xstrig()は，ジョイスティックのトリガーが押された回数を返す関数です。最初にsp_xinit()を実行してから，ジョイスティック1のトリガーAを3回押して，

print xstrig(1, 0)

を実行してみると，"3"と表示されるでしょう。その後，2回トリガーを押して，またxstrig(1, 0)の値を調べると，2になっているはずです。つまり，最初はsp_xinit()を実行してからのトリガー押下回数，その後は前回xstrig()を実行してからのトリガー押下回数を返すわけです。

この関数の便利なところは，トリガーを押すという「動作」が簡単にチェックできることです。標準のstrig()関数では，トリガーが押されているかいないかという「状態」しかチェックできませんから，押すという動作があったことを判定するには，「押されていない状態」→「押されている状態」という2回のチェックが必要になります。これは面倒臭いうえに，BASICの遅さのせいでチェック間隔が長くなってしまうと，トリガーを押したことを見逃す可能性も大きくなってしまいます。これではシューティングゲームなどでは連射が効かず，操作性が台なしです。xstrig()なら，割り込み処理のときにトリガーの状態をチェックしてトリガー押下を判定していますから，見逃しはまずありません。それでも10MHz機でサンプルゲームxsp.basを実行すると連射が効かなくなるんですが，strig()を使っていたら弾の出る数はさらに減って，半分以下になるはずです。

次に，scrpri()関数を説明しましょう。これはスプライト・テキスト・グラフィックの表示優先順位を変える関数です。X68000には表示優先順位を変える機能がちゃんとあるのに，X-BASICではなぜかこの機能がサポートされていないんですよね。これでは画面設計がものすごく制限されてしまうんで，この関数を作ってみました。

ほかにも説明していない関数がありますが，すべてこれまでに説明した機能に関連したものなので，リファレンスを見てもらえばわかるでしょう。あとは実際に使ってみて理解してください。

*

XSPRITE.FNCを作ってみて，こんな機能もあるとよかったなあと思う部分もあります。まず，スライドが直線的な動きだけじゃなくて，S-OSのYGCS（シューティングゲームコアシステム）みたいに三角関数移動なんかができるといいなあと思いますし，スプライトパターンのアニメーションの機能をつけて，たとえばキャラクタがトコトコ歩くというような動きができるようになっていたらとも思います。それに，BGのマップモードなんか，もっと本格的に取り組んだら，RALLY-X（一応説明しておくと，大昔のナムコのゲームです）が一瞬にして作れるようなものができるんじゃないかと妄想したりもしています。

ま，今回はごく基本的な機能だけで許してもらいましょう。これだけでも，X-BASIC上でのゲーム作りの可能性はかなり広がったんじゃないでしょうか。

先月もいったとおり，XSPRITE.FNCの狙いはX68000らしいピコピコゲーム作りにあります。X-BASICで，ビギナーの方も面倒臭がりの方も気楽な気分でゲームを作ってみましょう。あなたの個性がひと味違うピコピコゲームを生み出します（か?）。

BASIC拡張のすすめ

外部関数/ライブラリ作成の手引き

Asakura Yuji 朝倉 祐二

外部関数による拡張性はX-BASICの大きな特徴です
ここでは実際に外部関数を作成してみましょう
コンパイル時のライブラリとの共用についても考えます

発表当初は処理速度の点で不満の多かったX-BASICですが，XCの発売以来開発言語としての地位が高まった経緯はいまさらいうまでもないことでしょう。私もこれまでX-BASIC，XC(正確にはGCC)を使ってPENJANG!!，PUSH BON!などのゲームを作成してきました。が，そのたびにX-BASICに垂直帰線期間を検出するコマンドがないことに不満を感じてもいました。それでもX-BASICでゲームを開発してこれたのは，自分の必要としているコマンドを外部関数にしてX-BASICに追加することができたからです。

外部関数の作成はアセンブラを知っていても，書き方に決まりごとが多いので面倒に感じてしまう人もいますが，実際はそう面倒でもありません。

以下ではある程度アセンブラが使える人なら誰でも外部関数の作成ができるように，できる限り詳しく外部関数とライブラリの作り方を解説していきます。

外部関数とは

func〜endfuncで定義する関数に対して，外部関数はBASIC.CNFで，

FUNC=*.FNC

のように外部関数プログラムのファイル名を定義しておきます。X-BASICは起動時にBASIC.CNFに記述されている外部関数ファイルをX-BASICに組み込んでいきますので，外部関数で定義されたコマンドはあたかもX-BASICが最初から備えているコマンドのように使うことができます。

外部関数にする利点は，いちいちfunc〜endfuncを記述する必要がないのでプログラムを小さくできるということと，インタプリタで実行する場合も外部関数部分はアセンブラで記述されているため高速処理が可能になるということです。また，アセンブラで記述するということで，ハードに密着した処理を行うことができます。

外部関数の作成

外部関数は前述したように，*.FNCの外部関数ファイルに処理プログラムを記述します。ひとつの外部関数ファイル内に複数個の外部関数を定義することができるように，外部関数ファイルの先頭にインフォメーションテーブルがあります。

インフォメーションテーブルは外部関数全体の情報を定義するテーブルです。さらに外部関数名やパラメータの種類などを定義するテーブルがこのほかに4つあります。

リスト1に外部関数のプログラム例を掲載しますので参考にしてください。

1)　インフォメーションテーブル

インフォメーションテーブルは64バイトの大きさがあります。先頭の24バイトはX-BASIC上で外部関数を使うために，X-BASIC起動時の外部関数初期化ルーチンのアドレスや，ブレイクキーが押されたときに実行するサブルーチンの先頭アドレスを定義します（リスト1のコメントを参照)。普通に外部関数を作成する場合は，呼び出すサブルーチンのアドレスにrts命令を書きます。

続く8バイトは外部関数の将来の拡張を考えての予備領域に割り当てられています。呼び出すサブルーチンのアドレスにはrts命令を書いておきます。

続く12バイトは，トークンテーブル，パラメータテーブル，実行アドレステーブルの3つのテーブルの先頭アドレスを指定します。これらの各テーブルの詳細についてはあとで説明します。

最後に予備の領域が20バイトありますが，すべて0埋めしておきます。

2)　トークンテーブル

トークンテーブルにはX-BASICが外部関数を呼び出すときの関数名を定義します。関数名を0で区切ることによって，複数の関数名を定義できます。すべての関数名を定義したら，トークンテーブルの終わりを示す0を最後に付加します。たとえば，tcls()，tpattern()という2つの関数名をトークンテーブルに定義するには，

```
dc.b 'tcls',0
```

外部関数の集大成を

8ビット時代，BASICというものはマシンを操る唯一の手段として装備されていました。それゆえ，搭載されたハードウェアのすべては(まがりなりにも)，BASICから制御することができたわけです。

しかし，10月号および今回のテキスト画面用関数を見てもわかるように，X-BASICではX68000のハードウェアのごく一部にしかアクセスができません。テキスト画面は文字表示とマウスのみ，グラフィック画面もハードウェアに搭載された多彩な機能を駆使することはできません。一応，ひととおりアクセスできるスプライト関係にしても，X-BASIC ver.1時点ではBGに対する操作はできなかったのです。

特殊プライオリティや半透明モードなどはX-BASICでも使いでのありそうな機能なのですし，限定された使い方ならラスタースクロールなどの制御関数も不可能ではないでしょう。

外部関数はユーザーのアイデア次第でどんどん拡張していくことができます。理論上はほとんど不可能な処理はありません。

しかし，考えてみると意外と基本的なものが揃っていないのも事実です。6ボタンスティックやアナログスティックに対応したものとか，メモリやデバイスに対してももっと直接的に働きかけるような関数は必要です。SCSIやRS-232Cなどへのアクセスもあったほうがよいでしょう。

そのほか，X-BASICの不備を拡張できるような関数とか，ほかのシステムと連係したものも考えられます。グラフィックファイルの対応を増やしたり，いくらX-BASICが遅いとはいってもSLASHを制御するための外部関数があってもばちは当たりませんから。

そろそろX-BASICの集大成的なものをまとめる時期にきているのかもしれません。

```
    dc.b   'tpattern',0
    dc.b   0
```
となります。

3) パラメータテーブル

パラメータテーブルには外部関数が必要とするパラメータの数と種類を示すテーブル（パラメータIDテーブルと呼ぶ）の先頭アドレスを，トークンテーブルで定義した関数名の順番に関数の数だけ記述します。先の例だとトークンテーブルにtcls()，tpattern()の順番に2つの外部関数名を定義したので，パラメータテーブルは，

```
    dc.l   tcls_par
    dc.l   tpattern_par
```
となります。

4) パラメータIDテーブル

パラメータIDは1ワードでひとつのパラメータの型を表します。パラメータIDの数値の意味を表1に示します。

外部関数tcls()は引数はなし，tpattern()は，tpattern(x,y,dy,pat,plane)でx,y,dyがint型，patがchar型1次元配列，planeがchar型，tcls()は関数の戻り値はなし，tpattern()はint型の戻り値があるとすると，パラメータIDテーブルは，

```
    tcls_par:
        dc.l   void_ret
    tpattern_par:
        dc.l   int_val
        dc.l   int_val
        dc.l   int_val
        dc.l   ary1_c
        dc.l   char_val
        dc.l   int_ret
```
となります。

これを見てもわかるように，外部関数が引数を持たない場合でも，戻り値の有無を定義しますので，パラメータIDテーブルはひとつの関数に必ずひとつあります。

5) 実行アドレステーブル

トークンテーブルで記述した関数名の順番で実際の外部関数処理プログラムが置かれている実行アドレスを指定します。たとえば，

```
    dc.l   tcls_rtn
    dc.l   tpattern_rtn
```
となります。

パラメータの受け渡し

X-BASICから外部関数へのパラメータの受け渡しはスタックを介して行います。ひとつのパラメータは10バイトのエリアを必要とします。

X-BASICが外部関数を呼び出した直後のスタックの様子を表2に示します。ちなみに，表中のオフセットは外部関数が呼び出される直前のSPからのオフセットを表します。

表3に各パラメータの10バイトの内訳を示します。表3のなかで最初の2バイトはパラメータの型を表します。残りの8バイトはデータが格納されるエリアですが，パラメータIDテーブルで指定した属性により，表4に示したように有効なバイト数が変化します。表5にパラメータの構造，表6に配列ポインタの構造を示します。。

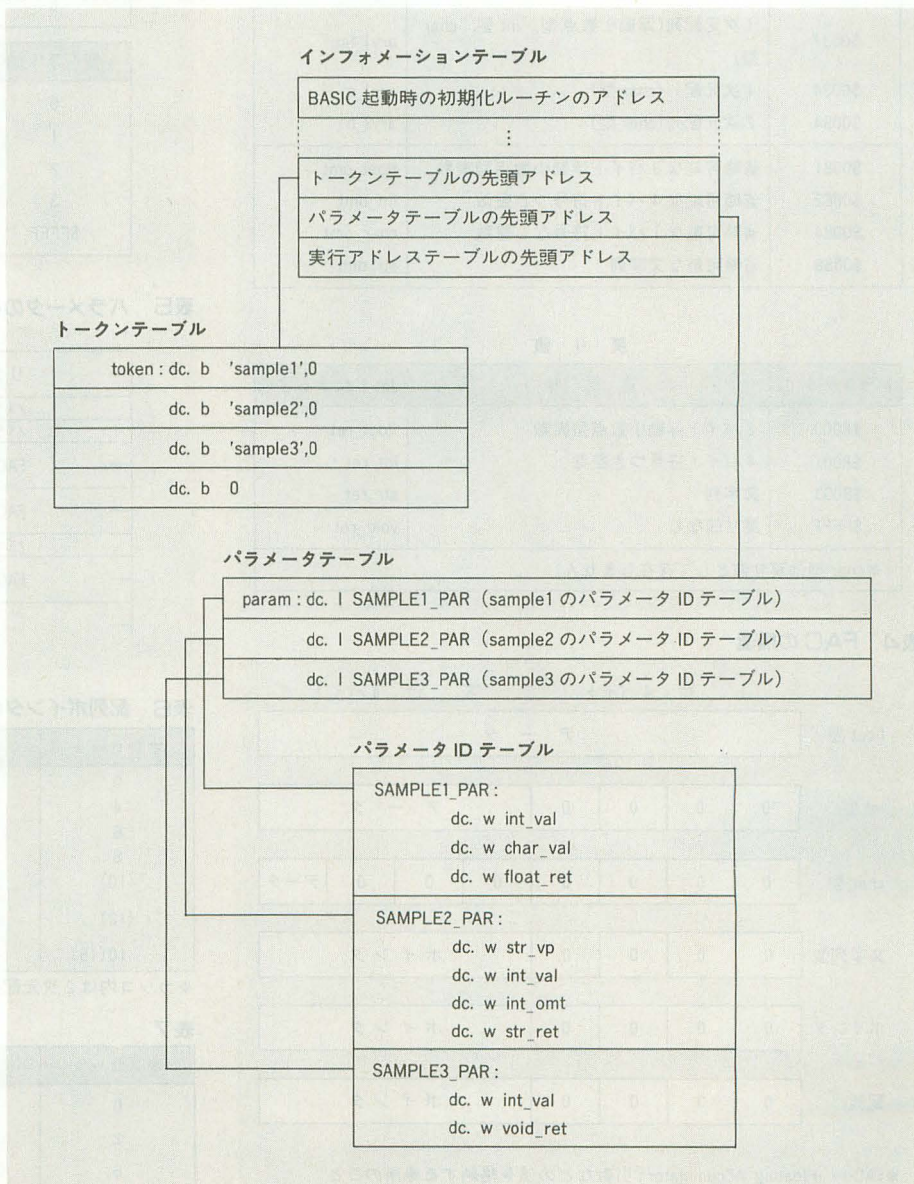
パラメータの受け渡しはX-BASICから外部関数への橋渡し部分であり，パラメータの受け渡しがうまくいかなければ，外部関数も期待どおりの動きをしてくれません。ここは重要なところなので，リスト1を例にパラメータの受け渡しの実際を説明します。

リスト1の外部関数が必要としている引数の型を表1を見ながら調べてください。どこを調べればいいかわからないという人は，もう一度パラメータIDテーブルの説明を読み直しましょう。

第1～3引数は4バイト符号つき整数(int_val)，第4引数は1次元配列（ary1_c），第5引数は1バイト符号なし整数(char_val)を必要とし，X-BASICへはint型の値を返す（int_ret）ということがわかったでしょうか。

最初に第1～3引数の取り込み（70～72行）から見ていきましょう。

外部関数が呼び出された直後は6バイト目以降10バイトおきにパラメータが格納されています（表2）。

図1　外部関数の構造

10バイトのうち先頭2バイトはパラメータの型を表しますが，いまはint型の変数であることがわかっていますので，ここは無視してデータが格納される残りの8バイトに注目します。表4を見るとint型データは下位4バイトに値が格納されることがわかります。これで第1パラメータであるint型データの格納されているアドレスを求めることができます。

第1パラメータの先頭アドレスがsp＋6，実際のデータが格納されているのはパラメータ先頭アドレス＋6ですから，つまりsp＋12のアドレスということになります。同様に第2～3パラメータも各パラメータ先頭アドレス＋6のアドレスにデータが格

表1

パラメータ

パラメータID	意　　味	fdef.hによる名前
$0001	8バイト浮動小数点型実数	float_val
$0002	4バイト符号つき整数	int_val
$0004	1バイト符号なし整数	char_val
$0008	文字列	str_val
$0011	浮動小数点型のデータ部のポインタ	float_vp
$0012	int型変数のデータ部のポインタ	int_vp
$0014	char型変数のデータ部のポインタ	char_vp
$0018	文字列型変数のデータ部のポインタ	str_vp
$003F	1次元配列(すべての型)	ary1
$0032	1次元配列(int型)	ary1_i
$0037	1次元配列(浮動小数点型，int型，char型)	ary1_fic
$0034	1次元配列(char型)	ary1_c
$0054	2次元配列(char型)	ary2_c
$0081	省略可能な8バイト浮動小数点型実数	float_omt
$0082	省略可能な4バイト符号つき整数	int_omt
$0084	省略可能な1バイト符号なし整数	char_omt
$0088	省略可能な文字列	str_omt

戻り値

パラメータID	意　　味	fdef.hによる名前
$8000	8バイト浮動小数点型実数	float_ret
$8001	4バイト符号つき整数	int_ret
$8003	文字列	str_ret
$FFFF	戻り値なし	void_ret

※char型は戻り値として存在しません。

表2

オフセット	サイズ(バイト)	内　容
4	2	パラメータの総個数
6	10	パラメータ1
16	10	パラメータ2
26	10	パラメータ3
⋮	⋮	⋮
(n−1)×10+6	10	パラメータn

表3

オフセット	サイズ(バイト)	内　容
0	2	パラメータの型
2	4	値の上位4バイト
6	4	値の下位4バイト

型を示す値	パラメータの型
0	8バイトの浮動小数点型実数(float型)
1	4バイトの符号つき整数(int型)
2	1バイトの符号なし整数(char型)
3	文字列型(str型)
$FFFF	省略された引数(void型)

表4　FACの構造

	上位4バイト				下位4バイト			
float型	データ							
int型	0	0	0	0	データ			
char型	0	0	0	0	0	0	0	データ
文字列型	0	0	0	0	ポインタ			
ポインタ	0	0	0	0	ポインタ			
配列	0	0	0	0	ポインタ			

※FAC……Floating ACcumulator：引数などの値を格納する場所のこと

表5　パラメータの構造

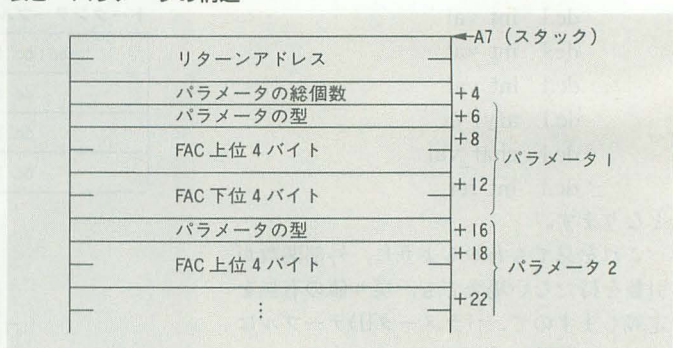



表6　配列ポインタの構造

オフセット	サイズ(バイト)	内　容
0	4	スキップオフセット
4	2	次元数−1
6	2	データのサイズ(1，4，8)
8	2	添字の大きさ(0～65535)
(10)	(2)	(2次元目の添字の大きさ)
(12)	(4)	(2次元の場合(?,0)の先頭のオフセット)
10(16)	各データのサイズ	データ領域

※カッコ内は2次元配列の場合

表7

オフセット	サイズ(バイト)	内　容
0	2	ワードで0を入れておく
2	4	戻り値の上位ロングワード
6	4	戻り値の下位ロングワード

納されています。

4番目のパラメータは配列ですので，データエリアの下位4バイトが配列へのポインタとなっています。1次元配列の場合，配列の内容は配列ポインタの先頭アドレス+10バイトの場所から格納されますから，73行で配列ポインタへのアドレスをa2.lレジスタに入れたものを，99行で10を加算して配列データの先頭アドレスを指すようにしています。

最後の第5パラメータはchar型です。データエリアの下位1バイトにデータがありますから，第5パラメータの先頭アドレス+9に実際のデータが格納されていることになります（75行）。

次に外部関数からX-BASICへ戻り値を返す場合の作法を説明します。

●エラーコード

外部関数が正常終了した場合はd0.lレジスタに0をセットしてプログラムを終了させます。

エラーがあるときは，d0.lレジスタに0以外の数値をセットして，a1.lレジスタに表示したいエラーメッセージの先頭アドレスを入れてプログラムを終了させます。エラーメッセージの最後にエンドコードの0を書くことを忘れないでください。

●戻り値を必要とする関数の場合

戻り値を必要とする関数の場合は，10バイトのリターンコードエリアが必要です（表7）。d0.lレジスタ（必要があればa1.lレジスタに）値をセットすると共に，リターンコードエリアにX-BASICに引き渡す戻り値をセットします。それからa0.lレジスタにリターンコードエリアの先頭アドレスを入れてプログラムを終了させます。

ライブラリの作成

新たに外部関数を作成した場合，その外部関数を使用したX-BASICプログラムをコンパイルするには，新たにライブラリを作成しなければなりません。ライブラリの作成方法についても簡単に説明しておきましょう。

大まかにいえば，外部関数プログラムからインフォメーションテーブルなどの各種テーブルと，パラメータのエラーチェックを取り除き，パラメータの受け取り方法を変更したものがライブラリになります。

パラメータの受け渡しは外部関数と同じくスタックを介して行いますが，ひとつのパラメータは型に関係なく4バイトとなっています。先の例でいえば，配列をパラメ

リスト1

```
 1: *------------------------------------------------------------------
 2: *
 3: *        X－BASIC用外部関数 テキストパターン
 4: *
 5: *                                        Programmed By Y.ASAKURA
 6: *
 7: *                                                    Version 1.00
 8: *
 9: *------------------------------------------------------------------
10:
11:         .include        iocscall.mac
12:         .include        doscall.mac
13:         .include        fdef.h
14:
15:         .text
16:         .even
17:
18: minx:   equ             0
19: miny:   equ             0
20: maxx:   equ             1023
21: maxy:   equ             1023
22:
23: * infomation table
24:
25:         .dc.l   x_init          * 普通は呼び先に"rts"を書いておく
26:         .dc.l   x_run           *       〃
27:         .dc.l   x_end           *       〃
28:         .dc.l   x_sys           *       〃
29:         .dc.l   x_brk           *       〃
30:         .dc.l   x_ctrl_d        *       〃
31:         .dc.l   x_res1          * 予備1(rtsさせる事)
32:         .dc.l   x_res2          * 予備2(rtsさせる事)
33:         .dc.l   ptr_token       * トークンテーブルの先頭アドレス
34:         .dc.l   ptr_param       * パラメタテーブルの先頭アドレス
35:         .dc.l   ptr_exec        * 実行アドレステーブルの先頭アドレス
36:         .dc.l   0,0,0,0,0       * 予備3 (0を入れておくこと)
37:
38: x_init:
39: x_run:
40: x_end:
41: x_sys:
42: x_brk:
43: x_ctrl_d:
44: x_res1:
45: x_res2:
46:         rts
47:
48: ptr_token:
49:         dc.b    'tpattern',0            * 外部関数名の定義
50:         dc.b    0                       * 終了コード
51:
52:         .even
53:
54: ptr_param:
55:         dc.l    tpattern_par            * パラメタIDテーブルの定義
56: tpattern_par:
57:         dc.w    int_val                 * X座標(int)
58:         dc.w    int_val                 * Y座標(int)
59:         dc.w    int_val                 * Y方向の表示方向と段数(int)
60:         dc.w    aryl_c                  * パタン格納された配列(char)
61:         dc.w    char_val                * パタン表示プレーン(char)
62:         dc.w    int_ret                 * 戻り値(int)
63: ptr_exec:
64:         dc.l    tpattern                * 実行アドレスの定義
65:
66:
67:         *=== テキストパタン ===*
68: tpattern:
69:         bsr     super                   * ユーザ→スーパバイザ
70:         move.l  12(sp),d0               * X座標
71:         move.l  22(sp),d1               * Y座標
72:         move.l  32(sp),d2               * 表示パタンの方向と段数
73:         movea.l 42(sp),a2               * 配列ポインタへのアドレス
74:         moveq.l #0,d3
75:         move.b  55(sp),d3               * パタン表示プレーン
76:         bsr     tpattern_main
77:         bsr     user                    * スーパバイザ→ユーザ
78:         lea.l   return(pc),a0           * 戻り値エリア先頭アドレス
79:         move.l  ret_low(pc),d0          * d0.l = 0なら正常終了
80:         rts
81:
82:         *=== テキストパタンプログラムスタート ===*
83: tpattern_main:
84:         bsr     check_x1y1
85:         bcs     tpattern_rts
86:         cmpi.w  #-1024,d2
87:         blt     dansuu_err              * 段数<-1024
88:         cmpi.w  #1024,d2
89:         bgt     dansuu_err              * 段数>-1024
90:         tst.w   d2
91:         beq     dansuu_err              * 段数=0
92: *       tst.b   d3
93: *       bmi     plane_err               * 符号なし整数なので負のチェック
は省略
94:         cmpi.b  #4,d3
95:         bcc     plane_err               * planeは0～3で指定
```

ータにした場合は外部関数と異なり，配列内容が格納されている先頭アドレスがスタックに入ります。

また外部関数ではSP以外のレジスタの退避/復帰を意識する必要はありませんが，ライブラリにする場合は，C言語とシステムでd3-d7およびa3-a7レジスタを使用しますので，必要ならばこれらのレジスタを退避/復帰する処理も追加します。

さらに外部関数名が“tpattern”だとすると，外部関数処理プログラムの入口に，関数名の前に“_”をつけて“_tpattern”というラベルを書きます。そして，

.xdef _tpattern

とラベル名を外部定義宣言します。ライブラリにするには，まずプログラムをアセンブルして＊.Oファイルを作成し，

lib /u ライブラリ名 ＊.O

とします。たとえばsample.sをライブラリにするなら，

as sample.s
lib /u sample.l sample.o

のようにします。

DEFファイル

X-BASICからC言語に変換するにはBC.Xを使いますが，新たに外部関数のライブラリを作成したなら，外部関数群をC言語に変換する際に参照するDEFファイルも作成しなければいけません。

詳しい説明はXCユーザーズマニュアルを見てもらうとして，ここでは1点だけ書きます。たとえば，引数に省略可能な文字列が指定可能な戻り値を持たない外部関数“test()”をDEFファイルに定義する場合，

test(S-)

のように引数の型記号の後ろに“-”をつけます。このようにしておくと，引数が省略されている外部関数をC言語にコンバートすると，省略されていた部分に'NASI'と4文字埋め込まれます。

先にライブラリはすべてのパラメータを4バイトで受け取ると書きました。DEFファイルを正しく定義していれば，省略可能なパラメータを省略した場合スタックに“NASI”の4文字が格納されます。このことはCユーザーズマニュアルに記述されていないと思うので紹介しておきます。

外部関数の場合，パラメータが省略されたかはパラメータエリアのパラメータ型を調べればわかりますが，X-BASICをCにコンバートして利用する場合，DEFファイルがないと省略可能な引数が省略されたか

```
 96:          moveq.l #0,d6
 97:          move.w  8(a2),d6
 98:          addq.w  #1,d6                * 添え字の大きさ＋1
 99:          lea.l   10(a2),a2
100:          move.w  d0,d4                * d0退避(d4=work)
101:          move.w  d1,d5                * d1退避(d5=work)
102:          lea.l   $e00000,a1
103:          lsr.w   #3,d0
104:          lsl.l   #7,d1
105:          adda.w  d0,a1
106:          adda.l  d1,a1
107:          move.w  d4,d0                * d0復帰
108:          move.w  d5,d1                * d1復帰
109:          add.w   d3,d3
110:          add.w   d3,d3
111:          lea.l   offset(pc),a0
112:          adda.l  (a0,d3.w),a1         * 書き込みアドレスを求める
113:
114:          *=== Y方向クリッピングチェック ===*
115:          move.w  #128,d5
116:          move.w  d2,d3
117:          neg.w   d3
118:          move.w  d2,d4
119:          bmi     dansuu_minus         * 下方向への表示
120:          neg.w   d5
121:          bra     dansuu_plus
122: dansuu_minus:
123:          neg.w   d4
124: dansuu_plus:
125:          cmp.w   d6,d2
126:          bge     dansuu_err
127:          moveq.l #0,d7
128:          divu.w  d4,d6
129:          subq.w  #1,d6
130:          move.w  d4,d2
131:          subq.w  #1,d3
132:          add.w   d1,d3
133:          tst.w   d5
134:          bpl     chk_maxy
135:          cmpi.w  #miny,d3
136:          bge     chk_maxy
137:          move.w  d1,d2
138:          subi.w  #miny,d2
139:          addq.w  #1,d2                * 調整後の段数
140:          move.w  #miny,d7
141:          add.w   d1,d7
142:          sub.w   d4,d7
143:          neg.w   d7                   * はみでたライン数
144:          subq.w  #1,d7
145:          bra     dy_ok
146: chk_maxy:
147:          cmpi.w  #maxy,d3
148:          ble     dy_ok
149: dy_err:
150:          move.w  #maxy,d2
151:          sub.w   d1,d2
152:          addq.w  #1,d2                * 調整後の段数
153:          move.w  #maxy,d7
154:          sub.w   d3,d7                * はみでたライン数
155:          neg.w   d7
156: dy_ok:
157:          movea.l a1,a3
158: loop_x:
159:          move.w  d2,d4
160:          subq.w  #1,d2
161:          lsr.w   #1,d2
162:          move.w  d2,d3                * d2(段数)退避
163:          *=== パタンをプット ===*
164: loop_x1:
165:          addq.w  #8,d0
166:          cmpi.w  #maxx,d0
167:          bgt     owari
168:          btst.l  #0,d4
169:          bne     loop_y2
170: loop_y1:
171:          move.b  (a2)+,(a1)
172:          adda.w  d5,a1                * 1ライン下のVRAM
173: loop_y2:
174:          move.b  (a2)+,(a1)
175:          adda.w  d5,a1                * 1ライン下のVRAM
176: loop_y_check:
177:          dbra    d2,loop_y1
178:          move.w  d3,d2                * d2復帰
179:          addq.l  #1,a3
180:          movea.l a3,a1                * 隣のバイトに移動
181:          adda.w  d7,a2
182:          dbra    d6,loop_x1
183:          swap    d6
184:          tst.w   d6
185:          ble     owari
186:          move.w  d6,d2
187:          moveq.l #0,d6
188:          bra     loop_x
189: owari:
190:          clr.l   ret_low
191: tpattern_rts:
192:          rts
193:
```

どうか判断する術はありません。ちなみに外部関数の場合，私が調べた範囲内では引数を省略するとパラメータエリア内のデータエリアには強制的に0がセットされるようです。

●インクルードファイル

新しく作成した関数のライブラリを使用するためにインクルードファイルBASIC.HまたはBASIC0.H内で関数宣言します。関数宣言についてはXCリファレンスマニュアルやいろいろなインクルードファイルを見れば理解できると思います。

サンプルの外部関数について

リスト1はX-BASICでテキストパターンを表示する外部関数です。HuBASICに存在したpatternコマンドを参考にしています。

tpattern(x,y,dy,pat,plane)

です。引数の型は，

int型：x,y,dy

char型1次元配列：pat

char型：plane

です。

x,y（0～1023指定）には表示するパターンの左上の座標を指定します。高速化のためx座標は外部関数で8ドット単位に切り下げします（例：xに10を指定→内部で8に変更）。

dy（−1024～1024）はパターンの大きさです。dyにマイナス値を設定すると下方向に，プラス値を指定すると上方向にパターンを描画します。patには描画するパターンデータをセットします。

plane（0～3）は描画するプレーンの指定です。おまけでサンプルリストを掲載しておきます(リスト2)。参考にしてください。

最後に

X-BASICはインタプリタの利点でデバッグは楽ですし，コンパイルすれば処理速度は劇的に向上しますし，プログラムを開発するのに非常に向いている言語だと思います。10月号の付録ディスクに収録されたXSPRITE.FNCのような外部関数が発表されると，改めてX-BASICが秘めている無限の可能性を感じます。

これからも素晴らしい外部関数が作成されることを願いつつ，今回の原稿が外部関数を作成しようという皆さんの役に立ってくれたら幸いです。

```
 194:         *=== d0(x1),d1(y1)がminx,miny,maxx,maxyに収まっているか調べる ==
=*
 195:         *=== 収まっていない場合 C=1 ===*
 196: check_x1y1:
 197:         cmpi.w  #minx,d0
 198:         blt     zahyou_err                  * x1<minx
 199:         cmpi.w  #maxx,d0
 200:         bgt     zahyou_err                  * x1>maxx
 201:         cmpi.w  #miny,d1
 202:         blt     zahyou_err                  * y1<miny
 203:         cmpi.w  #maxy,d1
 204:         bgt     zahyou_err                  * y1>maxy
 205:         andi.b  #$fe,ccr
 206:         rts
 207:
 208:         *=== ユーザ→スーパーバイザ ===*
 209: super:
 210:         clr.l   -(sp)
 211:         DOS     _SUPER
 212:         addq.l  #4,sp
 213:         move.l  d0,sspsave
 214:         rts
 215:
 216:         *=== スーパーバイザ→ユーザ ===*
 217: user:
 218:         move.l  sspsave(pc),-(sp)
 219:         DOS     _SUPER
 220:         addq.l  #4,sp
 221:         rts
 222:
 223: zahyou_err:
 224:         lea.l   errmes1(pc),a1
 225:         move.l  #1,ret_low
 226:         ori.b   #1,ccr
 227:         rts
 228: dansuu_err:
 229:         lea.l   errmes2(pc),a1
 230:         move.l  #2,ret_low
 231:         ori.b   #1,ccr
 232:         rts
 233: plane_err:
 234:         lea.l   errmes3(pc),a1
 235:         move.l  #3,ret_low
 236:         ori.b   #1,ccr
 237:         rts
 238:
 239:
 240:         .data
 241:
 242:         *=== 外部関数戻り値格納エリア ===*
 243: return:
 244:         dc.w    0
 245: ret_hi:
 246:         ds.l    1
 247: ret_low:
 248:         ds.l    1
 249:
 250:         *=== ワークエリア（定数） ===*
 251: offset:
 252:         dc.l    $00000,$20000,$40000,$60000
 253: maskdata:
 254:         dc.w    $7f7f,$bfbf,$dfdf,$efef
 255:         dc.w    $f7f7,$fbfb,$fdfd,$fefe
 256: errmes1:
 257:         dc.b    '座標値が範囲外です',0,0
 258: errmes2:
 259:         dc.b    '段数の指定に誤りがあります',0,0
 260: errmes3:
 261:         dc.b    'プレーン指定に誤りがあります',0,0
 262:
 263:         *=== ワークエリア（変数） ===*
 264: sspsave:
 265:         ds.l    1
 266:
 267:         .end
```

リスト2

```
  10 /*
  20 /* テキストグラフィック関数　サンプル
  30 /*
  40 screen 2,0,1,1
  50 /*
  60 dim char pat(31)={
  70    +255,255,255,255,255,255,255,255,&HAA,&H55,&HAA,&H55,&HAA,&H55,&HAA,&H55,
  80    +255,255,255,255,255,255,255,255,&HAA,&H55,&HAA,&H55,&HAA,&H55,&HAA,&H55
}
  90 int x,y
 100 /*
 110      /*=== テキストパタン ===*/
 120 for i=0 to 999
 130      x=rnd()*767:y=rnd()*511
 140      tpattern(x,y,8,pat,0)
 150      x=rnd()*767:y=rnd()*511
 160      tpattern(x,y,-4,pat,1)
 170 next
 180 end
 190 /*
```

ごめんなさいのコーナー(特別編)

10月号の付録ディスク「もみじ狩りPRO-68K」で次の不具合が見つかりました。

●ベル.LB

子ウィンドウを「取り消し」で終了した場合，以後子ウィンドウが開けなくなる。

●WIND.X

名称未設定のときにビットマップのプロパティ設定ウィンドウを開くとWIND.LBを開こうとする。

MAC.X（1993年10月号「秋祭りPRO-68K」などに収録）でリスト1，2(WIND.LZH，BELL.LZH）を入力後，それぞれ，2700バイト，885バイトでセーブしてください。

これらをLHA.Xで展開すると，WIND.BFD, CODE0080.BFD, CODE0081.BFDというファイルが得られます。差分を当てるにはBUP.X（1993年10月号「秋祭りPRO-68K」に収録）を使用してください。

●WIND.Xの場合

カレントディレクトリにWIND.XとWIND.BFDを置いた場合，

```
BUP WIND
```

で新しいWIND.Xが作成されます。

●ベル.LBの場合

まず，FSXを組み込んだうえで，

```
arlk -l ベル.LB >ind
arlk -e ベル.lb -iind
```

を実行します。これでリソースファイルがバラバラに抽出されたはずです。

ここに，先ほど入力して得られたファイルをLHA.Xで展開したもの（CODE0080.BFD, CODE0081.BFD）を置いて，

```
BUP CODE0080
BUP CODE0081
```

としてください。さらに，

```
arlk -a ベル.LB -iind
```

を実行することで新しいベル.LBが作成されます。これを従来のものと差し替えてください。

リスト1

```
000000  21 7B 2D 6C 68 35 2D 68 : 67
000008  0A 00 00 BE EC 00 00 9A : 4E
000010  1C 31 1D 20 01 08 77 69 : 73
000018  6E 64 2E 62 66 64 49 32 : A7
000020  48 00 00 05 AB 73 7C F0 : D7
000028  1A B7 1A FF 7D F7 83 0B : EC
000030  30 16 5F D6 C8 CC F0 A3 : A2
000038  1A 97 03 0E C3 01 90 84 : 9A
000040  6E C0 25 85 90 82 A4 15 : A3
000048  2C 86 5E 60 24 1B 30 10 : EF
000050  86 26 69 19 E6 79 66 19 : 0C
000058  C5 62 0E 44 84 86 22 F3 : 98
000060  31 D8 35 AC 86 43 0E 64 : 25
000068  24 0B 25 BD 27 60 38 ED : BD
000070  BA F5 DD 71 DD 6C B6 DB : D7
000078  64 E9 2F 59 24 96 4B 24 : FE
-----------------------------------------
CKSUM:  B9 03 54 09 3A 19 0F 40 AB51

000080  B2 CB 27 A5 B2 C9 2D B7 : A8
000088  A5 B6 96 D9 7A EB 69 65 : FD
000090  EB DB 2D DA F5 D6 F5 BA : 47
000098  D3 AA E4 60 03 70 02 EE : 24
0000A0  E0 D8 DA A3 01 B6 C9 6E : 23
0000A8  DB 21 64 61 64 66 1B 31 : D7
0000B0  D8 E6 59 0D 8F C0 91 80 : 84
0000B8  E3 77 7D F7 B5 B6 46 D2 : 51
0000C0  F7 FF 05 28 F9 FD 5F CC : 44
0000C8  F9 3B 3E B5 1D B7 EA 9E : 83
0000D0  2A 13 EF BA A5 FF FF 96 : 1F
0000D8  C9 97 49 ED BB 1E AF A4 : C2
0000E0  E8 FB 0E BB E1 75 1F 43 : 64
0000E8  EC A1 1F 6B EA FD 8F 80 : 0D
0000F0  BF 35 55 28 D4 A4 5F F2 : 3A
0000F8  94 9C F5 23 4F 85 15 1D : 4E
-----------------------------------------
CKSUM:  95 AD D4 B5 31 F8 61 2B 5F84

000100  22 8C 90 BF FC C9 7F 85 : C6
000108  1D 25 13 25 F9 CA 28 D3 : 38
000110  FB A4 ED 3F 74 3F 19 AF : 46
000118  F4 A2 3F 07 0E 8B EF F3 : 57
000120  1A FC 54 DB 7F 2B F9 16 : FE
000128  9D 32 14 69 E5 D4 51 7E : D4
000130  C1 0A 35 C2 2B FC 9A 79 : FC
000138  65 F8 6D 3A 7A 26 48 69 : 55
000140  C7 DB 78 DA 2D B6 33 5F : 69
000148  FA A3 51 7E 2A F8 1D AF : 5A
000150  EF BD E3 E8 99 35 C1 EB : F1
000158  F8 86 98 75 7F AB 4E E5 : E8
000160  46 9C E5 17 F9 D1 51 B5 : AE
000168  F9 73 CB F1 D9 6E 34 09 : AC
000170  96 E7 44 FD D7 66 AE DF : 88
000178  EA 76 9F FE 27 E4 C0 F2 : BA
-----------------------------------------
CKSUM:  72 54 B0 22 BF 95 2D DD 47D8

000180  E3 8E DC ED BF BA EE 9E : 3F
000188  1F E2 A3 C1 D5 E1 FD BF : D7
000190  5E CE B7 C3 5D F1 78 FF : 6B
000198  89 5A F4 5A 6D 41 65 86 : CA
0001A0  5B F3 DD 7C 72 A1 7A 06 : 3A
0001A8  B9 BE EA DB 4A 31 99 3C : 8C
0001B0  C4 AF 4F 57 B3 75 3B 5B : D7
0001B8  5D C5 9F B7 78 90 5E 92 : 70
0001C0  24 7C C2 8E 34 FE 1A 6B : A7
0001C8  17 FB 5B 57 10 A6 AC 91 : B7
0001D0  8C 48 09 C6 9F D3 79 F5 : 83
0001D8  E0 CB BD F5 76 4B 3A FC : 54
0001E0  8E 32 FD 67 B1 22 F7 3A : 28
0001E8  75 D3 7B AB 6F 66 6C A5 : 54
0001F0  62 93 D2 2E 94 F6 57 89 : 5F
0001F8  B6 74 27 01 C2 60 8F 70 : 73
-----------------------------------------
CKSUM:  E0 53 33 11 14 44 36 D6 FAE9

000200  AC 05 5B 4B EB 1A B6 43 : 55
000208  8F 4C 5D CD 1B 67 72 C7 : C0
000210  1F 8C 6F 0F 16 73 8C 18 : 56
000218  63 3B AC 3F 22 82 36 DA : 3D
000220  02 3E 9B B2 54 EA 8E 68 : C1
000228  86 4B 27 82 29 70 E1 90 : 84
000230  63 3D 4C 9D E2 17 72 68 : 5C
000238  C7 C3 FB 3A 0C 5B 89 13 : C2
000240  8A 1B E0 78 3B 6B B2 0D : 62
000248  E2 BD 26 30 DB 47 DF 39 : 2F
000250  32 83 84 03 A7 75 24 1F : 9B
000258  E2 D4 98 3A DE 4B 21 4C : 1E
000260  CD 63 23 17 04 BF 48 AF : 24
000268  81 7A CB 31 BE 26 A6 5D : DE
000270  6D 78 B1 6D D5 1B 86 47 : C0
000278  B3 39 70 B4 55 34 72 18 : 23
-----------------------------------------
CKSUM:  5D 5E 0D BF 30 E8 10 8B CEA6

000280  C4 94 84 E9 1A 48 C9 4B : 3B
000288  58 7C 48 72 0D 02 0C B1 : 5A
000290  5E D6 53 48 C8 B4 53 7C : 1A
000298  32 3E 33 79 90 41 9E 3F : CA
0002A0  91 61 40 C5 B9 4B EF C2 : AC
0002A8  C8 0A D0 0F D2 D1 CF 94 : B7
0002B0  5A C2 D2 4A D5 C0 4B 9D : B5
0002B8  21 49 7D 52 75 B4 2B 28 : B5
0002C0  F4 42 E6 BC 53 9F B5 15 : 94
0002C8  CD 69 DB 7C 0B 34 CF 37 : D2
0002D0  D9 CC 4C 22 35 39 53 58 : 2C
0002D8  D2 6B 16 9C 5B 2B 7C 93 : 84
0002E0  98 9E EB 55 4A 48 C2 59 : 23
0002E8  BD 8D A4 9F 96 5E 87 2E : 36
0002F0  39 77 16 68 49 80 C4 2B : E6
0002F8  E1 DF 90 48 87 B1 39 C9 : D2
-----------------------------------------
CKSUM:  5B FD 09 26 F2 DD 93 84 057C

000300  38 C9 27 5A 9A 6A D8 CC : 2A
000308  07 33 45 F7 1F 99 55 41 : C4
000310  71 00 44 AE 8D AE 5F A1 : 9E
000318  89 7C 2B 32 E5 F5 28 20 : 84
000320  58 A6 C8 DE A3 1C BB 97 : B5
000328  5A 88 ED B7 81 86 06 E1 : 74
000330  AA 4B 76 D6 D1 73 41 23 : E9
000338  46 AE 58 DE BF 69 27 0D : 86
000340  2B FA 98 A9 AB 99 D4 FE : 7C
000348  4A EF 19 AA F8 9A 63 95 : 86
000350  94 E4 6E 91 A0 66 70 F0 : DD
000358  44 90 99 6E 08 AA FA A0 : 27
000360  CF 11 50 68 32 35 75 D3 : 47
000368  D6 34 C5 E8 91 DB 96 33 : EC
000370  A4 84 BF 3E 89 83 77 32 : DA
000378  28 DD 2D D2 CB E1 F8 5F : 07
-----------------------------------------
CKSUM:  99 A2 17 2C 41 DB F8 30 FC97

000380  33 27 99 78 0F F7 6A 2F : 0A
000388  AD E0 E4 F6 A4 67 B9 7C : A7
000390  6A 31 CC 7F E6 65 20 67 : B8
000398  EC FF A3 B1 E3 CE 9D FB : 88
0003A0  FE 65 ED 9F 71 7C 4F 99 : C4
0003A8  05 BE 78 07 4C ED 29 2F : D3
0003B0  49 43 FB 35 5C 8F 18 3F : FE
0003B8  EA 9E FE 4C 69 9E 96 BA : 29
0003C0  99 A8 97 FF 00 1D FB 16 : 05
0003C8  AF 98 B8 37 F2 8B 13 D3 : 99
0003D0  8A 32 7B 11 5B 48 FB EC : D2
0003D8  B9 0E BA 0F 6E 8D 97 4B : 6D
0003E0  EF 68 FA 7F 8F EE FE 27 : 72
0003E8  D4 42 3E B2 9B 24 74 FF : 38
0003F0  71 09 D9 FC EE AB E2 F5 : BF
0003F8  27 CA F9 BD 2F 3F F6 FA : 05
-----------------------------------------
CKSUM:  52 38 D8 05 00 A0 F0 03 EB9F

000400  CF A6 CF E9 7D 7F 71 D1 : 6B
000408  7D DE 9B A2 D9 F5 9E C9 : CD
000410  08 F7 9D 7F 51 F2 1A F7 : 6F
000418  7B 58 C8 E8 34 A5 77 65 : 38
000420  5E A8 91 51 E9 6C 28 14 : 79
000428  53 AC 92 47 24 05 65 73 : D9
000430  B6 B6 A2 44 EC F6 B1 54 : 39
000438  9A E7 08 79 15 10 76 6F : 0C
000440  20 14 94 5D E8 5B 43 13 : BE
000448  B5 DE C3 BF AF B6 F5 5B : CA
000450  72 C0 45 A8 6D 2F B1 26 : 92
000458  FF FA 4D D1 0A FC 04 D6 : F7
000460  17 0D 75 C8 39 72 BA 44 : 0A
000468  6F A8 88 E6 21 63 FC C0 : C5
000470  A5 7D 3F C5 3A 0D 2A 59 : F0
000478  36 7C 3E 6E BC 6C 2D 87 : 3A
-----------------------------------------
CKSUM:  77 1E FF BD 47 0C 4E 8E C5DC

000480  5A 8B 22 3A 6F 3F BC 9F : 4A
000488  D4 B7 BB 41 E4 FE 26 B0 : 3F
000490  73 95 B2 DB B8 AC A6 C3 : 62
000498  2D B3 F3 B6 66 6E F2 AE : FD
0004A0  63 CA 66 C0 92 EC F9 33 : FD
0004A8  BB 2B 51 75 AA 30 03 8A : 13
0004B0  54 0D 92 03 2C D1 5D 1C : 6C
0004B8  5A 52 DD AD 3B 6C 23 61 : 61
0004C0  F5 81 C9 D6 E3 BA 03 61 : 16
0004C8  69 24 A5 A4 A1 DD F8 3D : 89
0004D0  3F 0B C1 EA C8 98 45 FF : 99
0004D8  4E 44 4D FC 70 25 CC 5F : 9B
0004E0  3A 59 CE 0D AA 86 24 62 : 24
0004E8  92 6D 6C 7C 4C 05 DA 5B : 6D
0004F0  89 FA 72 9F F1 73 8D 5D : E2
0004F8  80 5F 80 1B CB CF 75 88 : 11
-----------------------------------------
CKSUM:  5A F1 50 94 82 D1 02 98 DB95

000500  1F 5C 03 D9 BD 05 47 9C : FC
000508  D7 70 FA 41 62 46 EB 41 : 56
000510  61 9B DA 4C DA E1 4D 8F : B9
000518  FF C0 25 7F 6F 53 CD DE : D0
000520  9C 22 D8 6E A6 B9 C4 88 : AF
000528  4C 80 16 82 9F 75 D0 EF : 37
000530  F6 BE 67 AF 7A 35 37 F6 : A6
000538  9D 26 F8 7A E4 6A 20 71 : 14
000540  51 82 EA 79 11 B4 84 4A : C9
000548  C2 65 A1 4E F5 B2 FB F5 : AD
000550  1E BB DD 8D DB D3 A6 9A : 31
000558  CD 42 59 31 00 63 83 6A : E9
000560  F1 AD F9 0B 19 A3 11 81 : F0
000568  71 C1 5A 37 DC 32 0A 46 : 21
000570  43 C4 13 AD DB 7F AF 0C : DC
000578  42 46 16 75 BD FC 93 37 : 96
-----------------------------------------
CKSUM:  B6 09 86 E7 79 38 3C 75 8EF3

000580  DB 99 FA FB 7D AF 33 2E : F6
000588  3C 67 54 00 97 DF 97 2D : 31
000590  CC BB 6B B7 7E 97 93 D6 : 27
000598  E2 5E B5 BF BE EC CB F4 : 1D
0005A0  5B AC 0F 40 39 A9 A9 67 : 48
0005A8  AE E6 3D 97 DB AB 68 FC : 52
0005B0  06 06 66 25 90 F5 0B DA : 01
0005B8  43 FE 87 7D EA 36 45 24 : CE
0005C0  DD EB BC 6D 50 89 2D 24 : 1B
0005C8  22 EB 58 6F DB A3 46 11 : A9
0005D0  17 D4 B0 F0 65 CB 31 45 : 31
0005D8  B1 1B 12 56 F1 CB 1F 12 : 21
0005E0  59 C8 79 7E 86 65 BB 76 : 34
0005E8  83 07 A0 A1 26 B6 31 B2 : 8A
0005F0  98 03 41 77 44 92 CA A9 : 9C
0005F8  63 CA 97 E7 BC F7 64 EC : AE
-----------------------------------------
CKSUM:  B5 10 6E 89 0B F6 66 CF 798F

000600  16 A0 E1 73 E1 2D 19 06 : 37
000608  BF 1C 54 BE DC 25 36 A4 : C8
000610  9B 23 3F C4 39 8F F3 06 : 82
000618  07 49 DF 12 BE B8 79 5C : 8C
000620  18 63 C5 92 97 42 22 B9 : 86
000628  00 4E 54 20 B4 67 92 BC : 2B
```

```
000630  89 6B 7A B4 92 B8 77 95 : 78
000638  D2 15 52 6A 2D C5 AC 16 : 57
000640  A7 9D 49 C9 C6 26 80 4E : 10
000648  91 C4 86 92 5F 5E 7E 60 : 08
000650  0C 6A F9 D4 92 FC C8 C0 : 59
000658  3D 3C 0D 2D 39 1A 53 E3 : 3C
000660  D4 EE 24 E5 F8 B3 8E CF : D3
000668  5E F5 45 D9 B3 F6 2A 34 : 78
000670  EC DA 63 2F 5E A5 FD C0 : 18
000678  44 4B F7 8B EE CA D0 3D : D6
-------------------------------------
CKSUM:  CD 68 D0 AB A5 71 30 7D 3B53

000680  30 22 71 51 6C 25 B9 4D : AB
000688  70 8B 01 86 EA 87 06 D1 : CA
000690  7F F0 01 72 62 AF 4E 2A : 6B
000698  4F 50 3C 79 CF 21 7F D8 : 9B
0006A0  CB 26 A5 89 0D 6E A6 DF : 1F
0006A8  99 C7 AF 16 05 6C 37 EB : B8
0006B0  1A 97 E9 22 BF 48 86 45 : 8E
0006B8  75 0E 34 C5 77 3A 47 AC : 20
0006C0  94 41 62 D6 6C 99 3E 39 : 89
0006C8  35 72 31 03 D9 7D A9 7F : 59
0006D0  D6 D6 C6 48 8B A3 82 71 : DB
0006D8  27 75 5F 6D F0 48 87 0F : 36
0006E0  3A BA 77 F7 08 91 44 6D : AC
0006E8  73 6F 90 C6 DA 12 5C E1 : 61
0006F0  17 E4 74 1C 9E F5 D1 2B : 1A
0006F8  88 B0 BE 5A 38 6A 36 85 : AD
-------------------------------------
CKSUM:  73 3A 11 09 47 DB CD 11 0507

000700  32 9D EB 73 39 F6 93 FB : EA
000708  3B 17 5A CF 19 1C 0B 06 : C1
000710  91 05 D5 A1 5E 95 B3 6C : 1E
000718  95 0E 7B 88 81 8E 87 5E : 9A
000720  A2 2B 90 DE 2A 0F 30 05 : A9
000728  39 21 15 53 19 2E 1C 62 : 87
000730  30 76 E8 35 D5 A0 FE 9D : D3
000738  ED 5D 2B 34 D4 AD F3 25 : 42
000740  9A 28 EE 76 ED 32 03 34 : 7C
000748  4A 02 E6 EF 49 D2 7E 2B : E5
000750  60 27 75 F3 DD EB FB 64 : 16
000758  88 5E DB D8 9E 7E 9B 4D : 9D
000760  67 DE 82 28 78 98 FB A3 : 9D
000768  E0 6E A2 2C 73 27 A7 2A : 87
000770  FF 15 02 62 5A CA B5 04 : 55
000778  E1 35 BD 91 A1 D5 E0 6E : 28
-------------------------------------
CKSUM:  7E 2B 54 7C B4 8A 63 43 2B52

000780  00 C2 6B 59 A5 6D 81 08 : 21
000788  9E 68 B4 CF 07 D4 D3 7F : B6
000790  91 FA 40 45 01 5A 08 B9 : 2C
000798  AD 9E DC 1B 74 1A 23 E4 : D7
0007A0  02 BA FA 0F D5 37 49 C2 : DC
0007A8  4B ED 5D 09 36 9F 45 C3 : 7B
0007B0  C2 BA DC 0A 12 4F 72 4A : 7F
0007B8  62 BA B0 CC 88 3C A4 13 : 13
0007C0  10 30 92 6B 0B 66 2A B8 : 90
0007C8  E9 2F 21 32 3D 75 8E 1C : C7
0007D0  77 5E E2 32 74 87 96 D0 : 4A
0007D8  D2 9F 05 E3 42 52 0B 0D : 05
0007E0  CE 79 09 72 F6 73 58 02 : 85
0007E8  2D A3 99 6D 1D 79 22 F7 : 85
0007F0  E6 69 C9 33 1F 98 9D 8F : 2E
0007F8  FB E2 34 F0 FD 6F 07 5B : CF
-------------------------------------
CKSUM:  6B A0 57 2A F3 BD 9A 9A DEB8

000800  CE 6B 96 7D E2 B8 DC BD : 7F
000808  97 EE DF D1 39 1B 66 63 : 52
000810  84 B7 96 3D 86 03 E6 97 : 14
000818  16 35 4C AC E3 3B D4 E3 : 18
000820  04 B7 38 27 26 68 ED 78 : 0D
000828  C9 4D D2 E6 50 B2 E4 F1 : A5
000830  6B 78 8F 5F BC C8 0A F8 : 57
000838  78 F3 BB 34 21 DE 61 9A : 54
000840  6C 49 D9 6E 82 0B DA 7F : E2
000848  98 6A 54 CA 7B F4 F3 96 : 18
000850  3D 60 56 04 24 67 88 2F : 39
000858  CC 8D BA 7F 2C 88 9A 4C : 2C
000860  86 11 EC 5B 40 BF C6 39 : DC
000868  F1 8E 6A CC F0 11 2B BB : 9C
000870  32 95 4B DF 26 32 58 57 : F8
000878  D1 3F C9 68 35 E7 95 DF : D1
-------------------------------------
CKSUM:  36 C7 52 00 AF A8 05 4F 51F4

000880  A5 DE 3C A7 BB B6 9F 0B : 81
000888  78 1D AA 89 53 1E 95 64 : 32
000890  2F 63 C2 6A AC CB 47 80 : FC
000898  98 7C 6A 5C C4 04 4D 66 : 55
0008A0  B9 ED CC CB DA 89 25 39 : FE
0008A8  58 FC DE 36 3A 6E 86 85 : 1B
0008B0  42 51 7C 66 3D BB F6 B0 : 13
0008B8  B4 3F 32 AE 86 F5 9C D3 : BD
0008C0  D7 FA 6C 9A DC 8D 8A 96 : 60
0008C8  71 25 BF 7C 21 9F AC BC : F9
0008D0  66 08 F3 4B B7 1B B3 F6 : 27
0008D8  45 D8 DB CC 90 39 9A AD : D4
0008E0  22 7F 78 A5 A1 63 32 BE : B2
0008E8  30 61 D1 E8 94 26 93 35 : CC
0008F0  73 8C B6 CD 12 7E 76 E0 : 68
0008F8  24 14 47 00 90 E6 81 26 : 9C
-------------------------------------
CKSUM:  C7 D2 A9 92 70 B7 44 84 85EF

000900  F0 0F 47 70 0B 7B A4 78 : 58
000908  76 E0 81 67 93 8B 8D 8B : 74
000910  4D 70 95 BA AE 27 19 CC : C6
000918  4D 1B 55 CF D4 9E A9 44 : EB
000920  2E A3 A7 81 D8 9A 83 06 : F4
000928  35 54 DA E7 33 CD 40 6D : F7
000930  86 3B 66 19 34 81 56 87 : D2
000938  A4 56 86 43 57 52 8F C7 : C2
000940  E9 1F 99 DE 89 8D 97 89 : B5
000948  35 67 95 44 93 C8 53 93 : B6
000950  80 C3 F6 E5 7A A4 98 74 : 48
000958  FE C8 40 EE C5 63 53 06 : 75
000960  62 4F C6 C6 85 86 45 67 : F4
000968  BE 29 E2 72 2D 16 B7 86 : BB
000970  04 6E FE EE D7 08 A0 9F : 7C
000978  CF 40 CD 13 1D 0F 10 FB : 26
-------------------------------------
CKSUM:  1C 39 F6 52 B7 14 1C F1 2EC3

000980  C0 54 1F 5E 2B 52 66 2E : A2
000988  1B DA 9C 91 D3 3D 55 74 : FB
000990  78 49 29 3E 42 35 BE CF : 2C
000998  08 8F 29 E3 9E 74 D8 3E : CB
0009A0  F9 DF 22 C3 4B 97 53 12 : 04
0009A8  EA 9B 2A CB 03 31 76 E1 : 05
0009B0  5E 10 8C A6 D8 8D 6C E9 : 5A
0009B8  C1 CB 88 95 B9 89 AB 19 : AF
0009C0  24 B8 39 05 7E 2F 5A 53 : 74
0009C8  2E 53 52 3F C2 6B 49 26 : AE
0009D0  ED E9 91 D6 E0 00 10 C0 : ED
0009D8  1B 4B CD 3F CA D2 DA 00 : E8
0009E0  5A 38 84 FD 74 25 A9 FA : 4F
0009E8  87 4E 28 8D A7 15 4A B8 : 48
0009F0  42 66 57 F8 AF 52 68 35 : 95
0009F8  FF 4E 15 44 E1 DF 7D 9F : 82
-------------------------------------
CKSUM:  D9 D4 6E F8 52 ED 96 63 AD1A

000A00  A0 E0 D6 92 57 3C 20 A3 : 3E
000A08  14 DF ED A6 13 53 A7 39 : CC
000A10  4F D7 E9 91 BB B9 8C C3 : 63
000A18  9B 31 5C 0E BC 60 89 46 : 21
000A20  AB EA 3C 32 69 10 78 A4 : 98
000A28  61 77 10 7B 92 08 C0 8D : 4A
000A30  1A 82 8C 08 40 89 16 F9 : 08
000A38  47 0C C2 4F 3E EB 4F 49 : 25
000A40  F9 AD 30 AA 53 B9 ED 59 : D2
000A48  BE 3B 1E 18 A4 95 A0 0C : 14
000A50  85 A6 94 5C ED F5 01 37 : 35
000A58  CE 92 EF 2B 0C B7 12 82 : D1
000A60  B8 E2 48 DD 14 9B DA 21 : 69
000A68  44 76 48 F6 B7 9E 83 A9 : 79
000A70  47 3E 8E 81 1E E5 0C 97 : 3A
000A78  F2 5D 6A 3E 1A 3A 14 7D : DC
-------------------------------------
CKSUM:  4A C9 FB B6 4D 86 96 54 0875

000A80  14 7B 3A 8F F6 CB 77 C5 : 55
000A88  DC 7F 80 00 00 00 00 00 : DB
000A90  00 00 00 00 00 00 00 00 : 00
000A98  00 00 00 00 00 00 00 00 : 00
000AA0  00 00 00 00 00 00 00 00 : 00
000AA8  00 00 00 00 00 00 00 00 : 00
000AB0  00 00 00 00 00 00 00 00 : 00
000AB8  00 00 00 00 00 00 00 00 : 00
000AC0  00 00 00 00 00 00 00 00 : 00
000AC8  00 00 00 00 00 00 00 00 : 00
000AD0  00 00 00 00 00 00 00 00 : 00
000AD8  00 00 00 00 00 00 00 00 : 00
000AE0  00 00 00 00 00 00 00 00 : 00
000AE8  00 00 00 00 00 00 00 00 : 00
000AF0  00 00 00 00 00 00 00 00 : 00
000AF8  00 00 00 00 00 00 00 00 : 00
-------------------------------------
CKSUM:  F0 FA BA 8F F6 CB 77 C5 8407
```

リスト2

```
000000  25 3A 2D 6C 68 35 2D E2 : A4
000008  01 00 00 2B 07 00 00 A2 : D5
000010  25 41 1D 20 01 0C 63 6F : 82
000018  64 65 30 30 38 31 2E 62 : 22
000020  66 64 5D 0D 48 00 00 01 : 7D
000028  73 62 83 DA B4 AF 7F 6F : 83
000030  F2 4E D8 A4 4F 10 9A C2 : 77
000038  E3 C1 B3 C1 84 F1 FB 70 : F8
000040  C6 13 CD 7B 7A 69 51 4C : A1
000048  A6 55 3D C1 84 F6 B8 2D : 58
000050  70 44 0C 11 51 3C 13 FF : 70
000058  B8 EF 41 FC 57 05 C1 04 : 05
000060  6D EC A2 82 E7 85 11 10 : 0A
000068  44 1C 28 22 28 3B 91 05 : A3
000070  1C 39 12 E1 9A 0C 11 88 : 87
000078  C5 ED EC 57 04 B1 82 24 : 50
-------------------------------------
CKSUM:  83 7E 04 58 CA 3F E4 34 B89E

000080  88 42 08 B8 AE 48 2E 4B : F9
000088  96 E4 B9 6B 96 ED B8 6E : 47
000090  1A D8 B6 48 D3 68 D8 32 : 35
000098  0E 5D B5 87 32 B6 B5 EF : 33
0000A0  1C 37 57 BB 54 01 A6 D8 : 38
0000A8  03 51 1A 6D 53 EA B7 55 : 24
0000B0  4F 9D CA 3F AF 51 D8 00 : CD
0000B8  A5 AC F2 5A 4F DC 4E E8 : FE
0000C0  C7 56 E6 3A 06 3D 99 EC : 05
0000C8  EC 54 87 D9 54 FD 36 DD : 04
0000D0  87 E9 38 FC 31 D0 F7 8F : 2B
0000D8  DC 7D 7B FF E5 F3 1B C4 : 8A
0000E0  FD C4 4F 6C 14 71 46 12 : 59
0000E8  21 BC 20 65 46 92 0D AE : F5
0000F0  34 E0 08 6D 86 0B 34 59 : A7
0000F8  58 B3 92 2C FC E7 C3 E1 : 50
-------------------------------------
CKSUM:  19 4F 82 2B 3A 5D 21 05 40ED

000100  0C 23 18 91 46 70 21 11 : C0
000108  EA 45 47 0A 34 08 40 CE : CA
000110  A2 10 52 1E FB C6 25 8D : 95
000118  CB ED 48 57 11 3C DD 0C : 8D
000120  59 6E 78 DE 1F 04 55 9C : 31
000128  E8 BC 90 AB 16 4E D8 27 : 42
000130  D8 D8 0A 83 6C 5B AE 91 : 43
000138  7C 07 82 E0 2D 0A 8B 44 : EB
000140  5B EE 3F 52 DB DF C7 FC : 57
000148  42 E0 05 FA C7 3F 24 6F : BA
000150  C7 80 D0 C8 17 E9 CD A1 : 4D
000158  29 50 47 08 6C 86 D4 67 : F5
000160  26 9C E8 5C F3 77 24 D2 : 66
000168  DE F1 52 B8 C4 E8 49 E4 : B2
000170  80 A9 FA 38 77 AD F5 7F : F3
000178  B6 47 37 F2 CF F3 7A 7D : DF
-------------------------------------
CKSUM:  BF 89 53 56 76 BD 31 35 AB19

000180  FF AA 12 55 09 4C 7D 99 : 7B
000188  A4 33 FD 01 CE D9 B2 2F : 5D
000190  22 DA 5D 22 15 FA 8B F6 : 0B
000198  F0 6C 74 91 AE 82 5A FB : E6
0001A0  E4 34 D7 9C 75 F2 9D 65 : F4
0001A8  15 5F CA 73 E5 E6 D9 74 : C9
0001B0  97 DC E5 3C 65 07 01 BD : BE
0001B8  CF 66 1F 9E 09 46 D6 20 : 37
0001C0  D2 2B FB CB 5D 86 6C DF : F1
0001C8  F5 38 71 D8 E4 81 89 EF : 53
0001D0  17 57 DA A2 FE DA FB 8F : 4C
0001D8  4F 2D 9F 7F 28 99 04 D6 : 35
0001E0  26 48 13 A6 4E AA 97 6A : 20
0001E8  73 54 54 B2 C9 9E AB FE : DD
0001F0  F6 34 FD 8C E2 47 26 61 : 63
0001F8  31 C8 B7 11 B0 53 30 14 : 08
-------------------------------------
CKSUM:  01 77 85 AB 72 22 ED 7F 86A1

000200  BF 41 8B EC EE 6B F7 7C : 43
000208  40 25 0A 2D 6C 68 35 2D : D2
000210  44 01 00 00 EB 09 00 00 : 39
000218  A8 25 41 1D 20 01 0C 63 : BB
000220  6F 64 65 30 30 38 30 2E : 2E
000228  62 66 64 9F 72 48 00 00 : 85
000230  00 D0 5A 7B D6 25 65 FF : 04
000238  BC 41 16 3F C7 34 D7 81 : A5
000240  60 B0 47 5E B0 41 60 EC : F2
000248  3C AA 84 1C 56 08 82 C5 : 2B
000250  1E B5 58 AC 36 4E B1 44 : 50
000258  12 C2 E0 71 38 1F 80 AE : AA
000260  0E 41 DC E3 42 CD 5E 81 : FC
000268  07 2C 1C F5 27 5D 42 D6 : E0
000270  0A 11 EC 14 10 25 44 15 : A9
000278  82 62 A0 B1 D8 60 9C 50 : 59
-------------------------------------
CKSUM:  E5 18 96 F3 69 1B 37 19 57BD

000280  30 38 27 1B 8C 51 1B 6D : 0F
000288  BB 02 12 41 07 3E 14 33 : 9C
000290  1B 30 C1 CC 3A 4A 80 29 : 05
000298  2E 9F B2 8A 57 AA E3 3E : 2B
0002A0  7A AB FD DD 72 62 E0 80 : 33
0002A8  61 6F E9 6C 53 38 7D 1C : 49
0002B0  B2 34 88 42 01 5C 53 2B : 8B
0002B8  44 00 9C 5C 4B C6 41 B1 : 3F
0002C0  94 15 C5 2A BD F8 42 D0 : 5F
0002C8  75 5C 5C E6 B5 4E EB 55 : 56
0002D0  2F CA 7F E4 08 7F 01 6D : 51
0002D8  BA 3E 04 C7 D2 15 1F 68 : 31
0002E0  8C 7B 16 C7 D5 CF F1 56 : CF
0002E8  52 AD 43 DD C1 FE 4C 05 : 2F
0002F0  1B 11 2D A8 36 B6 3B AC : D4
0002F8  39 DA CA F0 60 14 B3 23 : 17
-------------------------------------
CKSUM:  29 E3 AA 90 AD B0 FB A3 A926

000300  50 8B 9B A8 18 91 18 79 : 58
000308  93 CE AA 80 BC 11 F6 EF : 3D
000310  63 CD DD EF 15 23 F6 94 : BE
000318  31 72 AA DF 9A 19 41 FA : 1A
000320  8C A9 5D 4A 8B A7 7A 01 : 89
000328  EF 4F 67 B7 9B A9 29 FE : C7
000330  E0 A9 50 AC B8 28 F7 15 : 71
000338  B0 4C FE 2C 8F 27 27 E6 : E9
000340  83 76 E6 A2 2C E2 12 3B : DC
000348  F1 89 8B BF 4B FE AE 17 : D2
000350  58 36 1C F8 D2 23 EA C9 : 4A
000358  D3 4D 3C 8D CD 08 B3 76 : E7
000360  B6 D3 4C 31 F6 69 2F BC : 50
000368  BC E5 E5 BC 49 AC D9 6F : 7F
000370  D6 59 64 00 00 00 00 00 : 93
000378  00 00 00 00 00 00 00 00 : 00
-------------------------------------
CKSUM:  69 18 3C A2 45 9D 6B AC 1263
```

DōGA CGアニメーション講座 ver.2.50 ……………………………(第19回)

マニュアルを手に再チャレンジ!

プロジェクトチームDōGA
かまた ゆたか

今回は，初めてCGAシステムのマニュアルを入手した人やバージョンアップをきっかけに再チャレンジする人向けに，挫折の原因のひとつであるモデリングについてマニュアルとは違う角度から具体的にせまります。

はじめに

8月号で募集した，バージョンアップおよび新規申し込みにたくさんのご応募ありがとうございました。申し込みが少ない場合は，採算が取れないので中止せざるを得ないなどといっておりましたが，おかげさまで予想した人数を上回り，ほっとしている次第です。

追加マニュアルの制作やバージョンアップ作業は「予想された範囲内で遅れている」といった状況です。発送作業も，トラブルがなければ，この号が発売される頃には半分ぐらいの方が最新バージョンを入手されていることと思います。なんとか今月中に全員に届くようにがんばりたいと思います。

さて，応募の振込用紙の通信欄にて，皆さんのご意見などを拝見させていただきました。新規申し込みの方からは「CGAは難しそうだと思っていたが，GENIEは驚くほど手軽で，CGAに興味が湧いた」というううれしい感想が多くありました。しかし，バージョンアップ希望の方のほうには「前回マニュアルを入手したものの，途中で挫折してしまった。今回のバージョンアップをきっかけに再チャレンジしたい」というパターンが目につきました。

新規申し込みの方からも「900ページのマニュアルと聞いただけでめまいがする。使いこなせるか心配」という不安の声も多いことですし，当連載ではしばらくの間，初心者向けの内容をお届けしたいと思います。

マニュアルのどこから読むの?

GENIEの場合は，できることが限定されているため，何をするのかが明確になっています。しかし，CGAシステムは，特に新規に申し込まれた方にとっては，いったい何から始めたらいいのか，皆目検討もつかないかもしれません。

ややこしいマニュアルをバインダーに収め，ハードディスクにCGAシステムをインストールしたあと「さて，この900ページのマニュアルを本当に読むの?」と呆然とされているのではないでしょうか。だからといって，「え〜い面倒だ。とりあえずプログラムをさわってみよう」というのは，だめです。だめも何も，まったく使えないでしょう。

もちろん，この900ページのすべてを読む必要はありません。大部分は，その都度，必要になったときに読むだけで，最初に読まなければならないのはほんの一部分です。ですから，まずマニュアルの読み方について解説します。

ずばり，第2章「CGA大学」(T-27)から始めてください。この「CGA大学」は，CGAの初心者を対象に，きわめて初歩的なことから，相当高度なことまで，系統だてて解説しています。

その名のとおり全体を大学にたとえ，「教養課程」「専門課程」「修士課程」「博士課程」の4段階にレベルアップしていきます。修士課程と博士課程は，一般の大学では「大学院」と呼ばれており，通常，大学を卒業したというためには，教養課程，専門課程をクリアすればよいのです。皆さんの当面の目標も，この2課程に置いてください。

「教養課程」は，「コンピュータ基礎概論」「CGAシステム基礎概論」「PES基礎実習」「CAD基礎実習」の4科目から成っています。

「コンピュータ基礎概論」は，大部分のOh!X読者には不要でしょう。わかりきった内容だと思います。

「CGAシステム基礎概論」は，非常に重要です。しっかり読んで理解してください。CG独自の考え方も多く，初心者にとっては直感的にわかりにくいのですが，このあたりの感覚はGENIEを使った皆さんには，ほとんど自然に身についていることでしょう。

「PES基礎実習」は，飛ばしてください。極端な話，これは修得しないほうがよいでしょう。PESというのは，ウインドウタイプのメニューシステムで，CGAシステムの各ツールをマウスで指定して起動したり，データファイルを種類ごとにまとめて表示してくれるので，確かに初心者にとってありがたいかもしれません。しかし，PESからは利用できないツールやオプションがある，いちいち起動するのが遅い，バッチファイルが使えない，などといったデメリットもあり，現在当チーム内で使用している者はひとりもおりません。そのため，バージョンアップがまったくなされておらず，ヘルプも前バージョンのままという完全に過去の遺物と化しています。キーボードに慣れていない方にはちょっと大変ですが，各ツールの起動はコマンドラインから行い，ファイルの表示はDSを使うのがよいと思います。

次の「CAD基礎実習」で，初めてCGAシステムを使うことになります。この「CAD基礎実習」が最初の難関

で，小学校の分数計算，高校の数列という感じで，落ちこぼれが大量に発生するところです。

しかし，安心してください。この「CAD基礎実習」は，20ページにわたって，たいへんていねいに解説していますので，気を引き締めてかかれば，大した問題もなくクリアできるでしょう。

第1の挫折

さて，大した問題もなくクリアできたはずなのに，ここで大きな問題が発生します。

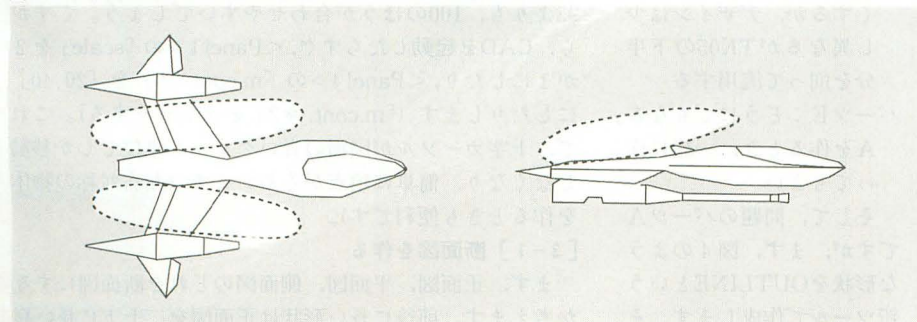
「どうして，たった4面から成る紙飛行機を作るのに，こんなに手間がかかるの？　この調子でオリジナルの物体を作るなんて，考えただけで気が遠くなる！」

ごもっともです。多くの方が挫折する理由のひとつです。しかし，それは大部分が誤解といえます。

まず，モデリングの作業は，最初から最後まですべてCADで行うわけではありません。現在では，いろいろなツールで形状を発生させたり，CADで簡単な形状を作ったあと，それらを組み合わせてモデリングするという手法が主流です。CADだけで物体を作るなんて，「TOSAKAじみた」考えです。CADをまったく使わなくてもモデリングできることは，GENIEですでに体験ずみですね。

次に，今回，紙飛行機を作るのに時間がかかったのは，CADの操作を修得するのに手間どったからであり，モデリングの作業自体は大したことはありません。試しにもう一度，紙飛行機を作ってみてください。半分の時間でできるはずです。さらに，10回同じ紙飛行機を作ったあとでもう一度行うと，1分以内で完成します。

つまり，慣れてさえしまえば，かなりの面数の物体でもサクサクと作れるのです。これはキーボード入力と同じだといえます。初めてキーボードで文章を入力しようとした人は，アルファベットを1字ずつ探して，変換して……これでは手で書いたほうが早いと思うでしょう。しかし，タッチタイピングなどができるようになれば，どんなに長い文章だって，無意識にどんどん入力できるようになるのです。

ですから，この第1の挫折は，一見「CADが大変だから挫折する」と思われがちですが，実は「CADに慣れる前に挫折している」というのが本質的な問題なのです。では，どうすればよいのか？　その答えは「CGA大学」の専門課程を卒業したあとに発生する第2の挫折と関連していますので，来月号でお話ししましょう。

モデリングの実際

ということで，今回はいろいろなツールを使った，より実践的なモデリングというものを具体的に紹介してみましょう。ツールの使い方から考え方まで，盗めるテクニックはどんどん盗んでください。

とはいっても，訓練もせずにいきなり実戦をすれば，討ち死にするのは目に見えています。とりあえず，最低限の特訓をしてからチャレンジしてください。

特訓1：教養課程「CAD基礎実習」の紙飛行機を5回作る

特訓2：修士課程「CAD応用実習」の紙飛行機2号を3回作る

特訓3：博士課程「CAD補講」を2回読む

ガンバレ！　コートのなかで泣くんじゃない！

[1]下絵を描く

今回のモデリングの題材として，GENIEのパーツを作ってみることにしましょう。7月号で制作した「OHX01」号が，どうもデザイン的に後ろが軽すぎるような気がしますので，ここにつけるブースター（？）用のパーツを作ってみます（図1）。うまくできるかな？

まず，どんなデザインにするか，紙に描いてみましょう。私が描いたデザインは図2です。

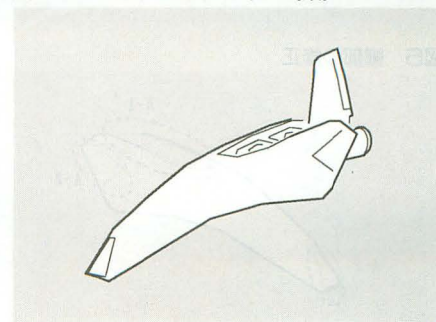
世の中には，下絵なんか描かずともいきなりCADを起動してサクサク作っていく人もいますが，そんな人はこの連載を読む必要はありませんね。

初心者の方は，図2のような透視図だけでなく，それを元にした3面図も描いておきましょう。とはいっても，透視図から，ちゃんと矛盾のない3面図を起こすなんて，ちょっとやそっとではできません。適当でいいのです。ただ，正面図と側面図で高さが全然違う，などという明らかな矛盾は避けておきましょう。

メカデザインというと，やたら細かく描き込む人がいますが，これはまったく無意味です。そんなに細部までモデリングできませんし，モデリングしている最中にどんどん変更されるに決まっているからです。細部よりも，全体のバランスのほうが重要です。

同様に，3面図だからといって，グラフ用紙に物差しで正確に描くというのも時間の無駄です。そんなに正確を期すなら，CAD上で作ったほうがましです。

図1　点線部に加えるパーツを作る

図2　ブースターのデザイン（案）

[2]方針を立てる

下絵ができても，いきなりCADを起動するのではなく，モデリングの方針を立てることが非常に大切です。まず，このデザインを，単純な形状の組み合わせに分解します。そして，個々の単純な形状をどうやって作るかを検討するのです。

図2の下絵を，私は頭のなかで図3のように分解しました。

単純な形状といっても，図形的に単純という意味ではなく，CGAシステムの各ツールで作るうえで単純ということです。ですから，どんなツール，どんな機能があるかということを，ひととおり知っておかなければ正しく分解できません。つまり，分解することと，個々をどうやってモデリングするかは，同時に考えながら進めます。「どうやってモデリングするか」には，次のような選択肢があります。

1）CADで1面ずつ作る
2）CADの機能（回転体など）を使って発生させる
3）CAD以外のツール（TAMENなど）で発生させる
4）過去のデータを流用（変形を含む）する
5）上記のデータを，ZANTEやMODELで一部を削る

また，次のような作戦も検討する価値があります。

6）複数の物体をKAMAで組み合わせる
7）作りやすい向きで作っておいて，あとで回転させる
8）単純な形状をBOXTRANSで変形させる
9）作りやすいデザインに変更する

図3　形状の分解

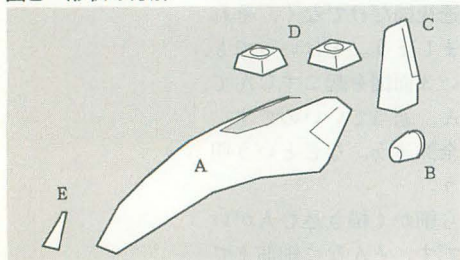



図4　OUTLINEで発生する図形

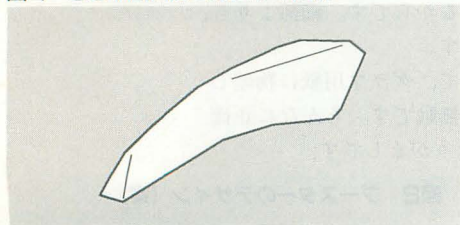
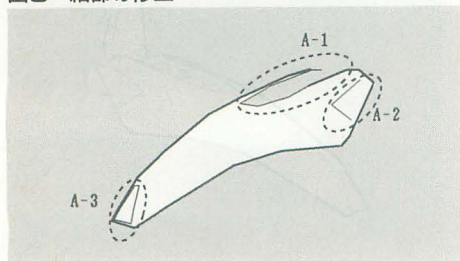



図5　細部の修正

私は，図3の各パーツを次のようにして作ろうと考えています。

パーツB：CADの回転体の機能を使うか，GENIEのEG02，EG07あたりを流用する

パーツC：平面なので，CADで簡単に作れる

パーツD：CADでがんばって作るか，さらに分解して台の部分と上部のレンズ状のパーツに分けて作ってKAMAでくっつける。面倒くさくなったら，GENIEのCN01を高くするか，デザインは少し異なるがTK05の下半分を削って流用する

パーツE：どうにでもなる。Aを作るとき，一緒に作ってもよい

そして，問題のパーツAですが，まず，図4のような形状をOUTLINEという新ツールで作成します。そして，図5のように，A-1の部分は面を削除することでくりぬきます。A-2の部分は面倒なので，デザインを変更して，なくしてしまっていいでしょう。A-3はCADでがんばって手作業で修正するつもりです。

どうでしょう。図2の下絵を見たときは，えらく大変そうな感じがしましたが，方針が立てば，手間がかかりそうなのはA-3ぐらいじゃありませんか。逆にいえば，方針が立たないような物体は作るべきではない（デザインを変更するべき）でしょう（方針が「努力と根性」だけってのはねぇ……）。

[3]OUTLINEを使う

では，方針どおり，まずパーツAをOUTLINEを使って作ってみましょう。このOUTLINEは，今回のバージョンアップによって加わった新作ツールです。大学1回生の新人・松村君が制作したので，多少バグがある可能性が高いですが，そんなことは気にしないように。

OUTLINEがどのようなツールであるか，という解説は省略します。そのためにマニュアルがあるんじゃないか，ということで，まずマニュアルをひととおり読んでください。なかなか便利そうなツールでしょう？

OUTLINEを使う前に，まずCADを起動して，断面図，平面図，側面図になる3つのポリゴンを作らなければなりません。この3つのポリゴンの良しあしで，Aのパーツがかっこよくなるかどうかが決まるので，慎重にいきましょう。

まず，念頭に置いておかなければならないのは，OUTLINEを使用すると，面数がかなり多くなってしまうということです。面数が多いと，作画に時間がかかるだけでなく，モデリングでも修正する面数が多くなり，作業量が増えます。面数を減らすことを最優先してください。

そのためには，アウトラインとなる3つのポリゴンの角数を思い切って減らします。3つのポリゴンの各角数がそれぞれ2倍になると，最終的に生成される面数は8倍（3乗）になってしまいます。多少の曲線など，大胆に直線で近似してください。

次に，最も大切なことは，平面図と側面図の「頂点がそろっている」ことです。「頂点がそろっている」の意味は，マニュアルの図を見てください。平面図，側面図の各頂点の位置に断面図を置いてつなげるので，各頂点のX座標をできるだけ同じにしておく必要があるのです。

X座標をそろえやすくするためにも，各頂点をわかりやすい数値にしておくのが有効です。たとえばX座標が93よりも，100のほうが合わせやすいでしょう。ですから，CADを起動したらすぐ，<Panel 1>の「scale」を2か1にしたり，<Panel 4>の「m.cont. 10」を「20,40」にしたりします（「m.cont. *2」をクリックする）。これで，十字カーソルが図面の青いグリッド単位でしか移動しなくなり，簡単に頂点がそろいます（左右対称の物体を作るときも便利です）。

[3-1] 断面図を作る

まず，正面図，平面図，側面図のどれを断面図にするか考えます。前後に長い形状は正面図を，上下に長い形

状は平面図を，左右に長い形状は側面図を選びます。また，面積がいちばん小さい，形状的にシンプルになる面を選ぶという考え方もできます。このパーツAの場合，当然，正面図が適当です。

図6が私が作った断面図です(単なる台形)。初心者の方は図7のように頂点数を増やしてしまいがちですが，先に述べたような理由でやめておきましょう。最終的にスムースシェーディングをかけると，粗い面数でも結構ちゃんと曲面になるものです。

操作としては，最初にアトリビュートを用意します。この時点では，どのような名前でもよいので，「test」とでもしておきましょう。「Attribute Mode」の「Attribute登録」に入って，「test」と入力します。そして，画面右下の「no」と表示されているところを「test」に切り替えます。

なお，断面図はどのような大きさでもかまいません。正面図だけを拡大して1面図の状態で制作するのもよいでしょう。正面図の右上の□マークをクリックします。1面図の状態でもう一度□をクリックすると，もとの4面図に戻ります。

[3-2] 側面図，平面図

側面図と平面図は，先ほどの「頂点をそろえる」必要性から，両方を見ながら作成します。もうひとつ注意する点は，X座標の最大値，最小値を同じにすることです。上から見た長さは900なのに横から見たら1200などということでは「アウトラインが矛盾しています」というエラーになります。

いきなり一発では，思いどおりの形にはならないでしょう。何度も描いて微調整します。その際，1つ描いては1つ消すのではなく，色を変えて重ねて何度も描きます(写真1)。そして，たとえばこの辺のカーブは黄色で描いたポリゴンの値を活かして，後ろの形は紫のポリゴン，というようにして，最終的な形を決めていきます。

操作方法は，作画するポリゴンの色を変えるのは<Panel 4>の「pen color」をクリックです。一度描いたポリゴンの頂点に合わせるのはキーボードの「8」，不要になった面を削除するのは「2」と「3」のキーでポリゴンを選択して，「DEL」キーを押します。これらのキーボードの操作は早く覚えてください。

[3-3] OUTLINEの実行

3ポリゴンができたら，適当な名前でSAVEします。たとえば「a1」でもよいでしょう。

OUTLINEの実行は，コマンドラインから，

OUTLINE /N a1 /Oa2

とします。/Nオプションは，マニュアルにあるとおり面数を少なくするためのものです。/Gオプションもつけるべきだと思うかもしれませんが，修正を加えたあとでもう一度SHADEを実行するつもりなので，いまの段階で実行しておく必要はありません。

また，このような左右対称の物体は，半分だけのポリゴンを作っておいて，/Kオプションで鏡面コピーしながら生成するという手も考えられますが，半分だけ作ってもバランスがつかみにくいので，今回はやめておきましょう。

[3-4] 確認・修正

できたa2.sufは，

CAD a2

として，CADの透視図で確認するか，それでわかりにくければ，AUTOで作画してみましょう。

AUTO /A2 /G a2.SUF

とすれば，自動的に20フレームで回転するアニメーションが作れます。アニメーションの速度が速すぎるようなら「ROLL DOWN」，1コマずつ見るならスペースキーを使います。

初心者向きの内容からはちょっと逸脱しますが，このAUTOを使って作られるアニメーションは，画角が60度なので，大きすぎて，下絵と雰囲気が違ってくる場合があります。そういう場合は，A2.FSCをエディタで修正して，視点の位置を3倍ぐらい遠くして，画角を30度ぐらいにするとよいでしょう。これらは，フレームソースの文法がわかっていないといけません。

さて，一発で，ちゃんと思いどおりの形状になっていることはまずないでしょう(写真2)。上記のように，3つのポリゴンの平面図，側面図を微調整して，再びOUTLINEを実行してください。そして，納得がいくまでこの微調整と作画を繰り返します（写真3)。

図6　断面図

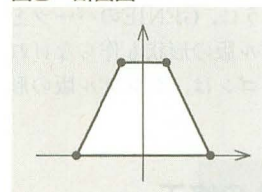



図7　断面図の悪い例

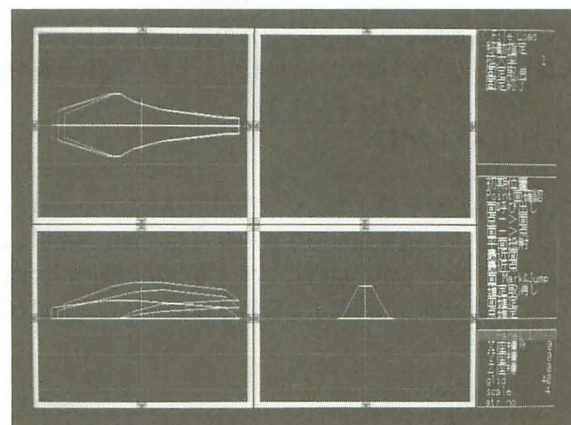
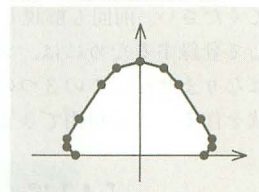



写真1　何度も重ねて描いてみる

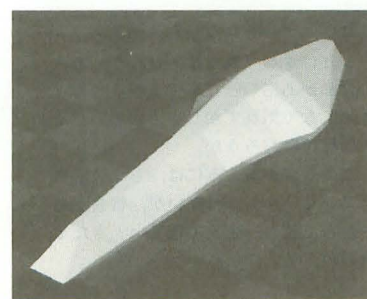



写真2　似ているが，ちょっとバランスがおかしいぞ

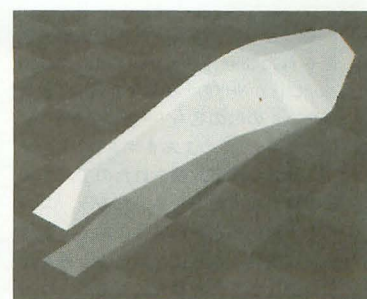



写真3　だいたいこんなもんかな？

図8　先が変になっている

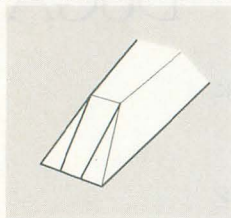



図9　3つの面を削除

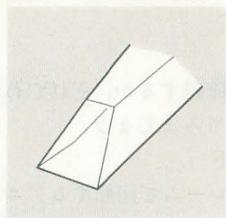



図10　加える面

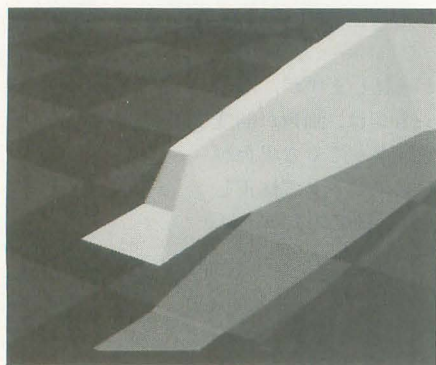
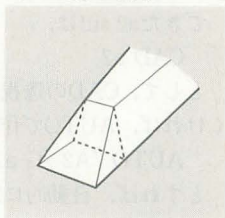



写真4　修正した先端部分

ひとつ注意することは，下絵にあまりこだわらないということです。厳密に下絵どおりの形状を作るのは難しいし，その必要はないはずです。生成された形状がそれなりにカッコよければ，それでよしとしましょう。モデリングしているうちに，どんどん当初のデザインからはずれ，まったく違うものができるのも楽しいものです。

また，OUTLINE実行時にオーバーライトして，もとの3つのポリゴンの形状ファイルをなくさないようにしてください。前回も解説したように，GENIEのパーツとして登録するためには，シンプル版の形状も作らなければなりません。この3つのポリゴンは，シンプル版の形状を作るときに利用できます。

[4]パーツAの修正

いちばん面倒そうな，パーツAの修正をしましょう。大部分は，CADの操作になります。千里の道も一歩から。地道にやっていきましょう。

[4-1] A-1の修正

　CAD a2

でCADにa2.sufを読みこんで，さっそくA-1のくりぬきをしましょう。透視図を回転させながら，立体を把握し，どの面を削除するかを検討します。もともと断面が台形なので，面が粗く，削除する面は2面しかないことがわかりました。

特定の面を削除するのは，以下のように操作します。まず，いちばん単純なのが，「Point面確認」と「面呼び出し」で削除する面を選択する方法です。これは，キーボードの「2」と「3」に割りふられていますので，キーボードから操作しましょう。「3」を押すと，順番に面が点滅しながら移っていきます。この点滅している面がポイント面です。SHIFTキーを押しながら「3」で，逆の順番で移っていきますから，選択したい面を行き過ぎた場合は，戻してください。「2」は，現在のポイント面を点滅して知らせます。この3つの操作で，ポイント面を削除したい面に合わせます。そして，DELキーで削除します。

このようなポイント面を移動させる操作は，必ず4面図の状態で行いましょう。なぜだかわからないのですが，1面図の状態では，このポイント面を知らせる点滅が非常に見にくくなるという欠点があるからです。

もうひとつの方法は，「最近点」(「8」キー)，「点－>面」(「4」キー)を使用します。削除したい面の近くに十字カーソルをもっていき，「8」キーで，十字カーソルを削除したい面のいずれかの頂点に一致させます。そこで「4」キーを押すと，その頂点を含む面にポイント面が移動します。削除したい面ではなく，その隣の面になってしまうことも多いのですが，再び「4」を押すと，その頂点を含むほかの面に順番に移動していくので，すぐに削除したい面にポイント面を設定できるでしょう。念のため「2」で確認したのち「DEL」で削除します。

この2つ目の方法は，操作はちょっと難しいのですが，面数が多い物体の場合，非常に有効です。

くりぬきたい部分の面をすべて削除したら，a3.sufとしてSAVEておきましょう。CADは終了せず，そのまま作業を続けます。

[4-2] A-3の修正

方針で決めたように，A-2の修正は省略します。A-3も面倒になってきたので，手を抜きましょうか。えっ？いい加減だって？そんなことありません。これもりっぱなテクニックです。

CADを使っていると，ついつい細かいところまで気になりますが，努力したところで，全体からみれば大差ないということがよくあります。むしろあまり細部にこだわると，完成する前に挫折してしまいます。モデリングは，小さいパーツの集合体として制作し，各パーツにはあまり時間をかけすぎないようにしましょう。とりあえず全部作って，つなげてみて，特に気になるところだけ修正するというのが正しいアプローチです。

A-3のような細部を修正する場合は，まず，その付近をクローズアップしましょう。「scale」を右クリックして1段階大きくします。さらに，各図面の左右上下にある△をクリックして，図面をずらします。このとき，FFEとは，ずれる方向が逆なので注意が必要です。透視図も，テンキーの「8」で近づけ，見やすい角度に回転させてください。

すると図8のように，A-3の部分の面が，不自然に3分割されていることに気づきました。これは，端が点や線になるときのOUTLINEのアルゴリズム上の限界です

CHUN-CHUN WORLDがみんなの歌に

第5・6回CGAコンテストで連続入選された佐野元さんの「CHUN-CHUN WORLD」がNHKの「みんなの歌」に登場した。その歌の名も「チュン・チュン・ワールド」(そのまんまやんけ)！

8月上旬から放映されたので，すでにご覧になった方も多いだろう。コンテストに出品された映像に加え，新作らしい映像，さらにはぬいぐるみを着た人間などとの実写合成もある。はっきりいって，オリジナルの作品のイメージとはかなり異なる。なんか，身内の作品があんあいった形で上映されると，なんとなく気恥ずかしいものを感じるのは私だけだろうか。

この号が出るころには，もう放送されていないかもしれないが，機会があったら，ぜひ一度見てほしい。きたぞ，ほらね，そこにいる！

(素直にバグといえって？)。よく見るとこれはマニュアルのT-307のT字接続になっていることですし，この3つの面などさっさと削除して，何も見なかったことにしましょう（図9）。

そして，図10のような面を1つ加えました。このように，すでに存在している頂点をつなげて面を作る場合，「8」キーの最近点をちゃんと使ってください。A-3の修正はこれだけにしておきます（写真4）。

ここで，a4.sufとしてセーブします。

［4-3］アトリビュートの設定

Aのパーツの形状はこれでよしとして，最後にアトリビュートを設定しましょう。前回GENIEについて解説したとおり，GENIEのパーツとして使用するためには，あらかじめ用意されたアトリビュート名を使用しなければいけません。今回は，全体を「BodyM」に，底面になる部分（図11）を「BodyD」に，そしてA-3の修正で加えた部分を「WinLg」にしてみました。

アトリビュートは，現在すべて「test」になっているはずです。まずこれを「BodyM」にします。「Attribute Mode」に入って，「登録名変更」をクリックします。「test」を選択したのち，「BodyM」と入力します。

次に，底面とA-3を変更するためには，まず「Attribute登録」に入り，「BodyD」と「WinLg」を入力します。そして，「面のatte.変更」に入ると，「BodyM」，「BodyD」，「WinLg」が並んでいます。そこで，もうそろそろ慣れてきた「2」と「3」を使い，底面となる面にポイント面を移しては「BodyD」をクリック，A-3の面にポイント面を移しては「WinLg」をクリックということを繰り返します。A-3の面とすべての底面を変更すれば，作業は終了です。

以上で，CADによる修正が終了しました。a5.sufとしてSAVEしたのち，終了してください（写真5）。

［5］SHADEで曲面化

パーツAに対してSHADEを実行し，曲面化させます。SHADEはバージョンアップによって，特定のアトリビュートだけを曲面化できるようになりました。

SHADE a5 /OA /ABodyM /ABodyD

として実行します。/Aオプションを2回指定しているのは，「BodyM」の部分と「BodyD」の部分の境界線を曲面化したくなかったからです。この機能もバージョンアップによるものです。詳しくは追加マニュアルの「SHADEのバージョンアップ」をご覧ください。

このSHADEを使うときの注意としては，できるだけ鋭角（90度以下の鋭い角）の部分を曲面化しないということがあります。「BodyM」と「BodyD」との境界線なども鋭角です。これはプログラムが悪いというより，尖った角を滑らかな曲面にすること自体に無理があるのです。強引に曲面化すると，黒いシミやしわのような影が発生する原因となります。

さて，出力されたA.sufがパーツAの完成品です（写真6）。作業自体は大した量ではありませんが，文章で説明すると，結構長くなりましたね。

おわりに

「はじめに」のなかに，「追加マニュアルの制作も予定された範囲内で遅れている」と書きましたが，この「おわりに」を書いている時点でもまだ追加マニュアルが完成していないので，「予定された範囲外で10日ばかり遅れている」に訂正いたします。ということで，今回は，本文以外のコラムを書いている余裕がありませんでした。お許しください。

マニュアルの発送は，この調子だと，10月中旬～11月中旬になると思われます。12月になっても届かない方は，振込用紙のコピーを添えて，お問い合わせください。

さて，今月はモデリングの話の途中で終わってしまいましたが，続きはちゃんと来月行います。といっても，パーツB，C，Dなんて30分もあればできちゃいますけど。まぁ，文章にすると結構長くなってしまうかも。皆さんも，文章を読んでいるだけではぴんとこないでしょうから，やはり実際にやってみてください。

ところで，次の「ある計画」のために，ペイントソフトで風景などの絵を描ける方を急募します。それぐらいなら協力できそうだという方は，至急，当チームまでご連絡ください。

問い合わせ先
〒533　大阪市東淀川区
淡路5-17-2 102号
DōGA内　担当・奥中

図11　底面の部分

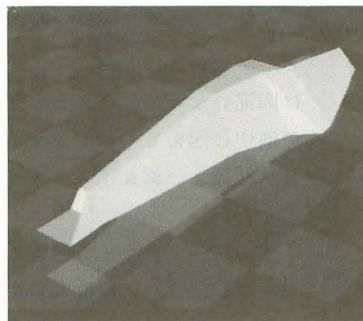
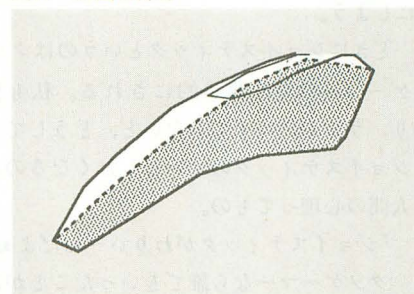



写真5　CADによる修正終了

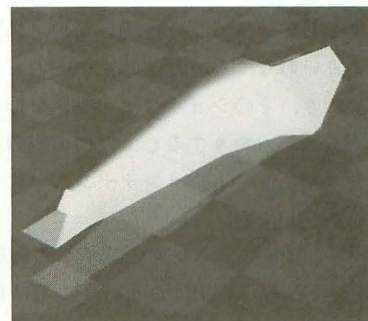



写真6　SHADEで曲面化する

宍戸氏プロ第1作発表

第4回CGAコンテストグランプリ作品「猿蟹合戦」の宍戸光太郎さんがプロとしての活動も始めたが，その記念すべき第1作が出た。その名も「宇宙少女刑事ブルマ」だ！　さぁ，君もレンタルビデオ屋を探してみよう。

タイトルがちょっと危ないが，18禁なんてことはないから，安心して借りられる。しかし，安心して家族の前で見るには，ちょっと問題があるぞ。日本の映像文化の明日について考えさせられる作品だ。

もちろん監督は宍戸さんではない。CG部分を担当しただけだ。宍戸さんに聞いたところ，ものすごく短い日数で相当量を作らされて苦労したそうだ。「ハリウッドのスペシャルエフェクトにもけっしてひけをとらない」というから必見だ。

宍戸さん，ごくろうさまです。

瀧流ジョイスティック周辺あれこれ

Taki Yasushi 瀧　康史

今回はアーケード用コンパネをそのまま使った本物指向のジョイスティックを作ってみましょう。6ボタン対応や多機種対応にするための方策についても解説します。

ジョイスティックのすすめ

ゲーマーに必需品なのがジョイスティック。ジョイパッドも悪くはないけど，ここではやっぱりジョイスティックということにしよう。

ときにジョイスティックというのはクソゲーマー*1の非難の的にされる。私も然り。うまくプレイできないと，どうしてもジョイスティックのせいにしたくなるのが人間の心理ってもの。

「ジョイスティックがわりぃ～んだよぉ」

クソゲーマーなら誰でもいったことがある台詞だろう。なんて自分が可愛い台詞。私もしょっちゅういっていたが，最近はあまりいわなくなった。外でゲームでもしない限り，そのジョイスティックを作ったのが自分だったりするし，仮に市販の部品を使ったにしても，メンテナンスをするのは自分だからだ。

「ジョイスティックの調子が悪いなぁ」

今度からいい訳はこういうことにしよう。

わがままな人間たちに，かわいそうにも非難の的にされるジョイスティックだからこそ，丈夫に，コントロールしやすく作らねばならない。

と，いうわけで，今月の連載では金に糸目をつけずにジョイスティックを作る。できるだけ安い部品というようなことは考えないが，やたら成金趣味にするつもりもない。「高い＝よいもの」という公式は，私の頭の中には成り立っていないからだ（といっても，同時に高いからやはりよいという公式も成り立っている）。

そういうわけで，瀧流ジョイスティック周辺あれこれである。

*1 クソゲームをする人ではなく，クソなゲーマーである。

どんなジョイスティックを買うか？

まず，ジョイスティックを作る。といっても市販品で優秀なものがあればそれを利用するのが正しい。それが高くっても。

どんなジョイスティックが気にいるかは，個人次第で，私が押しつけるものではない。だから個々で探してほしい……といいたいところだが，それではあまりにナンなので，「瀧流」と銘打った限り，偏見を含めて瀧流ジョイスティック論を論じたいと思う。

ジョイスティックは，ゲーマーのハートと，ゲームとのインタフェイスであることは間違いない。グググっときた私のハートを適切に伝えてくれるジョイスティックほど，私はよいジョイスティックであると評価したい。

このハートをうまく伝えるジョイスティックだが，これには個人差がある。私が凄く使いやすいと思っているジョイスティックでも，遊びにくる友人は見事にジョイスティックのせいにしてくれる。クソッ。もうお前には家でゲームをやらさん。といいたいところだが，彼にも並なみならぬ理由があるのだろう，きっと（半分はプライドだろうな）。

しかし，個人差はあってもたいていの人が気にいってくれるジョイスティックがある。それが，アミューズメントセンター（ゲームセンター）のゲーム用コンパネ（コントロールパネル）。これと同じものでやらせると，プライド高き友人も，あまりスティックそのものには文句はいわなくなる（スティックが滑るとか，高さが悪いとか環境に文句をいい始めるが）。

まぁ事実，使いやすい。

幸い，私はこのアーケードタイプのジョイスティックが非常に使いやすいと思っている。こういった趣味が，体のサイズのように*2アウトオブスタンダードになっていなかった事実，自分に感謝することにしよう。アウトオブスタンダードに走りすぎると，必ずどう考えても納得しがたい，余計ともいえる金が飛ぶからだ。編集部では中野氏が，モロにアウトオブスタンダードに走っている（と私は思う）。彼の趣味はゼビウススティックである。もはやゼビウススティックが使いやすいという人は少ないため，あまりこのタイプのジョイスティックは売っていないのが事実。別にゼビウススティックをこよなく愛することは悪いことではないので，彼には，ぜひゼビウススティックコンパチブルなスティックを探してもらうなり，普通のアーケードタイプ（ゲーセン）のジョイスティックに慣れるかしていただければ，今回スティックを作るほうとしても幸いである。

とりあえず自分のためにローテクをするのが基本である。自分のために努力できない者が，他人のために努力できるわけがないというのが私の持論であり極論*3。

市販のジョイスティックを探してみたところ，割合アーケードタイプに近づけているものが多い。ジョイスティックには，当然のごとく，ボタン部，スティック部がある。スティック部のカチカチというやつはマイクロスイッチを利用したもので，現在アーケードタイプのジョイスティックで主流になっているものである。

しかし，どうもアーケードタイプのボタ

ンとスティックの原価が高いのか，市販されているコンピュータ用，NEO･GEO用，メガドライブ用のジョイスティックは，この小型版を利用したり，ジョイパッドのようなゴムを利用したものが多い。

たとえば，NEO･GEOスティックは安いことは安いが，レバーは小型でフニャフニャしているし使いづらい。ボタンなんて連射できないほど腐っている。さすがにこれはいただけない。

そんなこんなで行き着く先は，結局，アーケードタイプのジョイスティックになる。そこで，まずゲーム基板などが売っている店に行ってコンパネを買うことにする。コンパネというのはジョイスティックのスティック，ボタン，そしてそれらを固定する金属板で構成されている。アーケードのゲーム筐体はそのままこれをつけることができるように作られているので，このまま売られているのだ。

このコンパネだが，秋葉原のメッセサンオーなどに行くと，うぢゃうぢゃ売られている。そのほか，ゲーム専門誌の通販広告などでもみつけることができるかもしれない（ただしコントロールボックスではない）。

では，コンパネを買う前に，どんなジョイスティックを作るか決めることにしよう。

まず，ボタン。X68000では小細工をしなければ，1Pにつき2ボタンしかつなげない。アーケードに2ボタンのものはそんなにないので，3ボタンスティックでも買って，ボタン配列を，A，B，A＋Bとでもしておけばよい。しかし，最近のX68000ではスト2なるものが発売されたので，やっぱりボタンは6ボタンがほしい。

6ボタンスティックのコンパネでもまがいモノはいっぱいあるようだ（編注：編集室ではセイミツの特殊コンパネ以外はこう呼ばれる）。ただ，まがいモノを買うと，1Pと2Pの部分が妙に短かったりして，2P側でプレイしていると，1P側の大キックを左手で押してしまったりする。下手するとボタンが水平に配置されてたりして，さらにやりにくさに磨きをかけてくれる。

探せば，まがいモノでも，スト2の配置とまったく同じものもある。しかし，写真撮影のときに美しくないので，とりあえず純正（?）ということで，いまをときめくヴァンパイアのコンパネを買うことにした。これは12,000円ぐらいだったかな?

ボタンとスティックは，とりあえずすべてついているので，あとはコンパネを固定する箱さえ購入すればよい。この編集部にはヘビーなユーザーが多いためか，ボタンも時期が過ぎると底が抜けてしまうが，そうなったらボタンだけそっくり換えればよい（どうも大パンチの具合がよくないなと思って中を見ると，ボタン内部の電極がプラスチック板を突き破っていたのには驚いた。いまは強化ボタンにしてある）。

箱のほうだが，メッセサンオーにこれにピッタリハマる箱が売られていた。あわせて2万円しなかったし，多分，自分で変な加工をするよりも安上がりなので，これを購入する。この世界では買ったほうが安いというのは，よくある話だろう。

このセットにはネジがついていないので，適当なネジを購入するとよい。ネジおたくではないので（そういう問題ではないような気もするが），ネジの名前が思い出せない。大きさを考えてネジ屋さんで各自悩んでいただきたい。

これでジョイスティックの心臓部は片がついたことになる。

＊2　身体のサイズは身長184.5cm，体重88kg。まだまだ安物を買えるサイズだが，そろそろやばくなってきた。別に腹が出ているので88kgもあるわけではない。でっぱらな服なら結構売っているのだ。

＊3　自分を愛せない者は他人を愛せないという持論もある。多少自分は自分の愛し方が足りないのではないかと思っているが。

ジョイスティックの機能

ジョイスティック自身の機能だが，下手にいろいろつけるよりも，役割分担したほうがよいと私自身は思っている。ジョイスティックに持たせる機能は，せめて連射，ボタンのカスタマイズぐらいだろう（今回はそれさえも分けた）。

このジョイスティックの機能だが，ジョイスティックそのものが，どうやってコンピュータと接続されているかを理解しないことには話は進まない。

まずは，図1を見てもらいたい。

図1はジョイスティックケーブルを挟んで，コンピュータ側とジョイスティック側に分かれている。したがって，R01はコンピュータ内部に実装されている抵抗ということになる。

順に話をしていこう。

ジョイスティックポートのようなポートを入力ポートという。理由はジョイスティックによって伝わってきたゲーマーのハートをパソコンに伝えるものだからだ。

入力ポートというのは，たとえば，図1の点Aの電圧を入力すると考えるとわかりやすい。ここではデジタル回路なので，簡単に電圧が高い状態をハイレベル，低い状態を，ローレベル（グランドレベル）ということにしよう。

ジョイスティックというのは，単なるスイッチの塊である。もっとも単純なジョイスティック回路は図2に示した回路で，こ

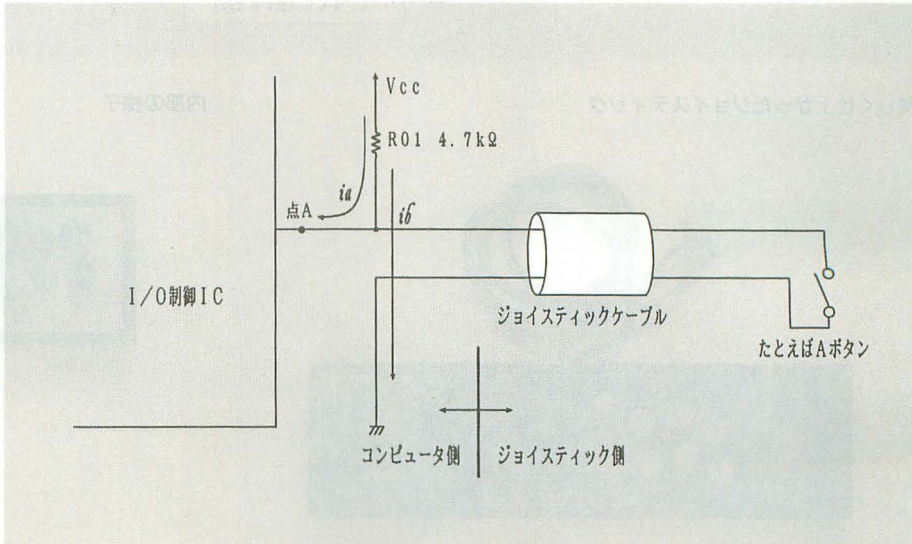
図1　ジョイスティックインタフェイス周りの回路

のまま作ればジョイスティックができあがる。ただ，図2のジョイスティックは実はNEO・GEOのジョイスティック改なので，このままでは当然X68000にはつながらない（これはあとで説明する）。

図2の回路を見ればわかると思うが，ジョイスティック中のすべてのボタン，スティックは単なるスイッチにすぎない。注目してほしいのはスイッチの両端だ。どのボタンも片側は，必ずGNDにつながっている。

これからわかるとおり，ジョイスティックは入力ポートを，単にGNDレベルに落とすものにすぎないことがわかる。これでなぜ動作するかが，図1の説明なのだ。

図1でボタンが離されているとき，点AはVccと抵抗R01を経てつながっている。当然のごとく，電流はiaしか流れず，ibには流れない。したがって，電圧レベルはハイレベルになる。抵抗R01を通る分だけ電圧降下が起きるが，これは流れ出る電流iaが小さいためほとんど無視でき，結果的にハイレベルを保つことになる。

もしこの抵抗がなく，かつVccにもつながれていなければ，点Aは電圧的に宙に浮いたかたちになるので，入力ポート自身が安定しなくなってしまうのだ。

このような役目を果たすR01のような抵抗をプルアップ抵抗と呼ぶ。プルアップ抵抗の形態はさまざまであるが，たいていのマシンのジョイスティック周りは原理的にこのようになっていると考えて差し支えない。

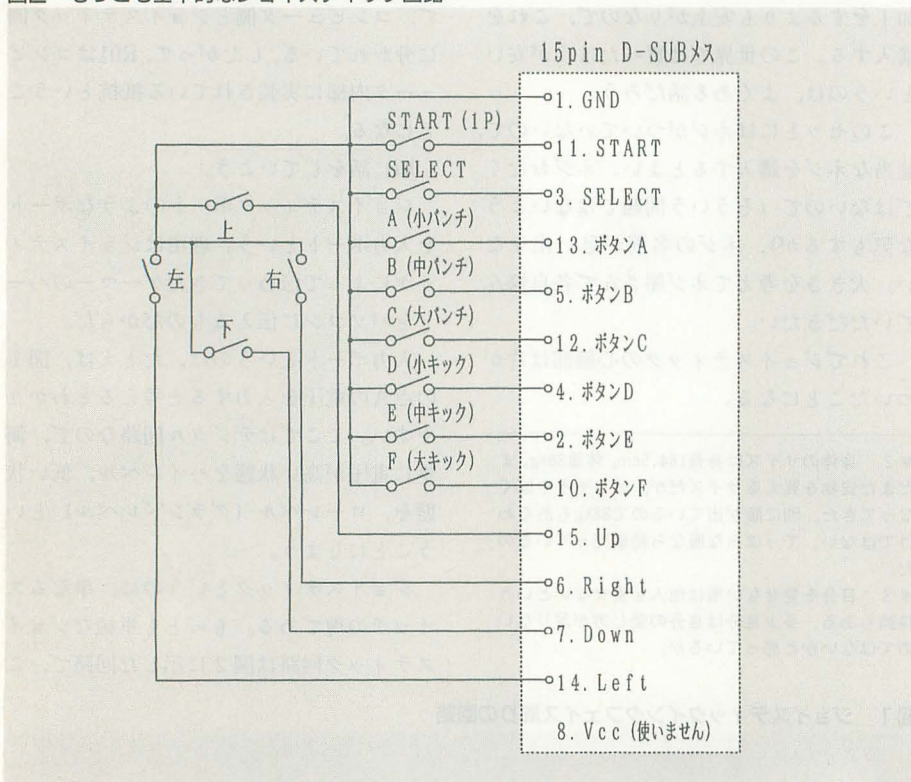
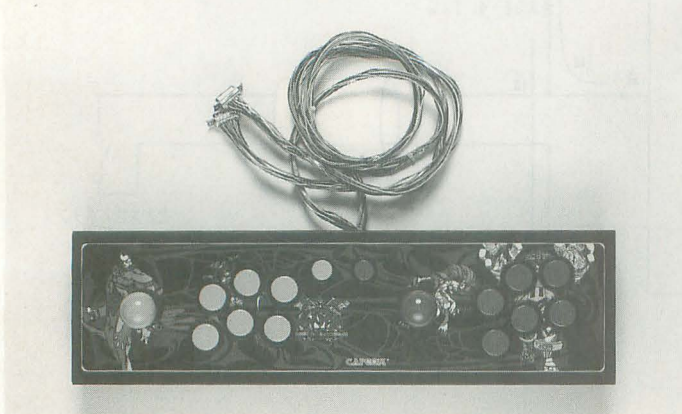
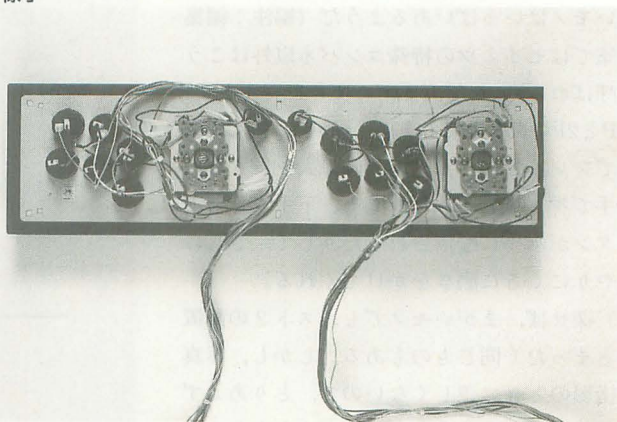
ここで，スイッチが押されたらどうなるだろう？ 当然のごとく，点AとGNDは「単なる線」で結線されることになるから，点Aは有無をいわずGNDレベルまで落ちる。プルアップ抵抗がそれなりの大きさでついているのは，このときにVccからGNDが短絡状態にしないためであると考えてよい。

iaはほとんど流れないため，ibはオームの法則より，Vcc/4.7k＝1.06mAとわずかだけ流れることが計算できると思う。向きもGND向きだということを忘れずに。

ジョイスティックを作るとき，ジョイスティックが単にGNDとのスイッチの塊であると考えすぎている人も多い。実際は図1の点Aの電圧を計るものだと考慮して製作しなくてはならない。こうしたほうが，電気的にも安定したジョイスティックが作

図2　もっとも基本的なジョイスティック回路

表1　NEO・GEOスティック配線

1.	COM(GND)
2.	ボタン　E (N.C)
3.	SEL
4.	ボタン　D
5.	ボタン　B
6.	Right
7.	Down
8.	+5V
9.	ボタン　D
10.	ボタン　F (N.C)
11.	START
12.	ボタン　C
13.	ボタン　A
14.	Left
15.	Up

美しく仕上がったジョイスティック

内部の様子

れることはいうまでもない。

ところで，ゲーマーの皆さんは，愛用のジョイスティックを作る際，X68000だけでなく，メガドライブや，NEO・GEOやPCエンジンなどといった家庭用ゲーム機にも使いたいと思わないだろうか？　愚問かもしれない。当然のごとく思っているだろう。

今回製作するジョイスティックはアダプタさえ作れば，どんなタイプのゲームマシンにも接続できる。回路図は図2のもの。スト2タイプの6ボタンジョイスティックの場合，セレクトボタンがないのでつなげられないが，それ以外はすべて図2の通りである。8のVccはコネクタまでは接続だけして，今回の回路では使わない。

実はこの15ピンというのは，ボタンE，Fを除いて，すべてNEO・GEOのボタン配列だ。したがって接続さえできれば（ボタンはかなり嫌な配列ではあるが），そのままNEO・GEOに接続できる。実際のNEO・GEOは，15pin D-SUBであっても多少奥の深いD-SUBなので，接続するのにはコツがいる。この奥の深いD-SUB15pinメスコネクタはときどきメッセサンオーに，NEO・GEO用のケーブルとして売られているので，うまく見つければ，なにも考えることなく接続することができるだろう。

手に入らなかったら，あのフニャフニャなNEO・GEO スティックを破壊してコード部分だけ使ってもよいかもしれない。

連射や，ボタンカスタマイズ機能の類をつけたいのも山々だが，スティック側にはあえてつけない。なぜなら，ジョイスティックは1本だけではないからだ。この文章の意味はコラムを読んでいただければわかるだろう。

図2の回路で作成したジョイスティックが写真1である。編集部で購入した，ヴァンパイアのコンパネには，スティックとA，B，Cボタンのコードがついていたので，加工はほとんどしなかった。

D，E，Fボタンは1Pも2Pもグランド以外のコードがついていないので，同じぐらいの長さ，同じぐらいの太さのコードを購入し接続する。

気をつけるべき点は，キックボタン（D，E，Fボタン）のグランドと，スティック，A，B，C，1Pボタンのグランドをつなぐこと。これらはちゃんとつなげないと動作しない。

また，2Pボタンが1P側のグランドに接続しているので，この2Pボタンのグランドを1P側からはずして，2P側につける。

別に1Pと2Pのグランドが接続していても構わないが，たとえば1Pはメガドライブにつけっぱなしで，2PにX68000をつなげたときのことを考えて，1Pと2Pは完全に分離しておく。GNDぐらい，別に接触してもどうってことないだろうが，これは気分の問題だ。

D-SUBへのハンダづけは，次の手順で行うとうまくいく。まず，コードを2mmぐらい剥き，それにハンダメッキをする。十分に温まったハンダゴテなら，1秒ぐらいでコードの先は暖かくなり，ハンダを当てると染み込むように流れ込む。これがハンダメッキの状態。すべて（1Pにつき12本）ハンダメッキしたら，D-SUBへハンダを流し込む。この作業は慣れればすぐに終わるだろう。下手にハンダが太かったりすると，失敗するので直径0.5mmぐらいのハンダを買っておくこと。

D-SUBにハンダを流し込んだら，あとは簡単だろう。接続すべきコードをD-SUBの接続箇所に当て，十分に温まったハンダゴテを当てれば，すぐに融けてコードとD-SUBコネクタは接続される。コード数が多いので慣れない人はつらいだろうが，頑張ってほしい。

これでできあがり。

なにも機能は持っていないジョイスティックなのに，なかなか壮観に感じてしまうのは，私だけだろうか？

参考までに表1に，拡張NEO・GEOスティック配線表を載せておこう。

ボタンカスタマイズ機能

できあがったジョイスティックをNEO・GEOなどに（なんらかの方法で）つけると，4ボタンを使うゲームのすべてがやりづらいと感じるだろう。4ボタン必要なゲームは，サムライスピリッツや餓狼伝説2／SPECIALなどがあるが，SNKもなにを考えているのか，この2系統はボタン配置が

図3　瀧流ジョイスティックの基本回路

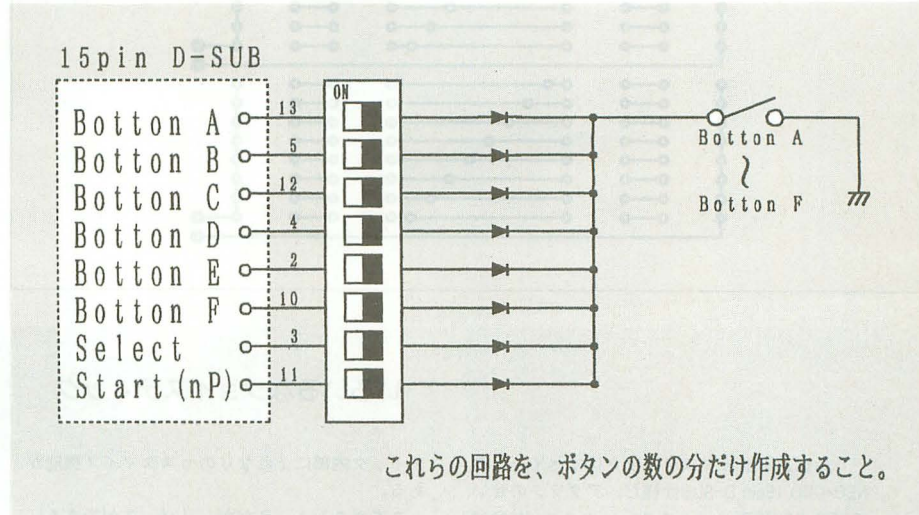
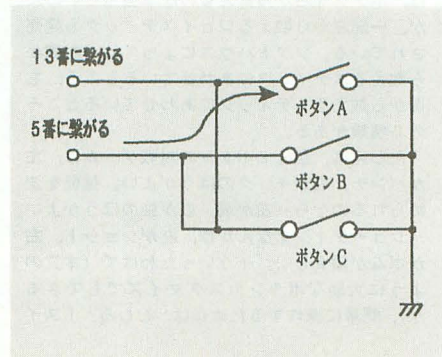
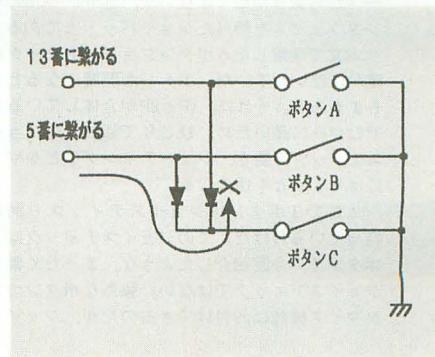



図4　同時ボタン押しの回路（誤り）

図5　同時ボタン押しの回路（正解）

まったく違う。多分，並なみならぬ理由があるのだろう。

私は基本的に便利なものを使ってしまう軟弱者なので，できるならば，サムライスピリッツはスト2コンパチブルなボタン配置でやりたいし，餓狼伝説も同じように，パンチが上，キックが下という配置でやりたい。そうなると，ボタンはゲームごとにカスタマイズできたほうがよい。

そこで，ボタンカスタマイズ機能である。

ボタンをカスタマイズする方法は，いろいろ考えた。たとえば，SRAMバックアップにでもして，現在のカスタマイズを記録する機能など。しかし，これらは回路が複雑になるだけだし，ちょっとしたときに変えられないという難点もある。

楽に，しかも安く作り上げるにはどうすればよいかと考えた挙句，作り出したのは，図3の回路である。

すべてのボタンをすべての機能で使えるように！ そういった思想から作られたのがこれである。ダイオードを利用した，単なるワイヤードORだったりするが，これで十分機能する。

回路の説明の前にまずは機能から説明しよう。図3の右手に，Button A～Button Fと書かれているが，これは実際のボタンである。これはすなわち，ボタンの数だけこの回路があることを意味している。

たとえば，サムスピのカスタマイズを例に出そう。サムスピは，4ボタンのNEO・GEOベースのゲームなので，以下のボタンの使い方をしている。

A　小斬り
B　中斬り
C　小蹴り
D　中蹴り

これに加えて，

A＋B　大斬り
C＋D　大蹴り

である。

私はスト2のボタン配置に慣れているの

図6　プリントパターン裏面（原寸）

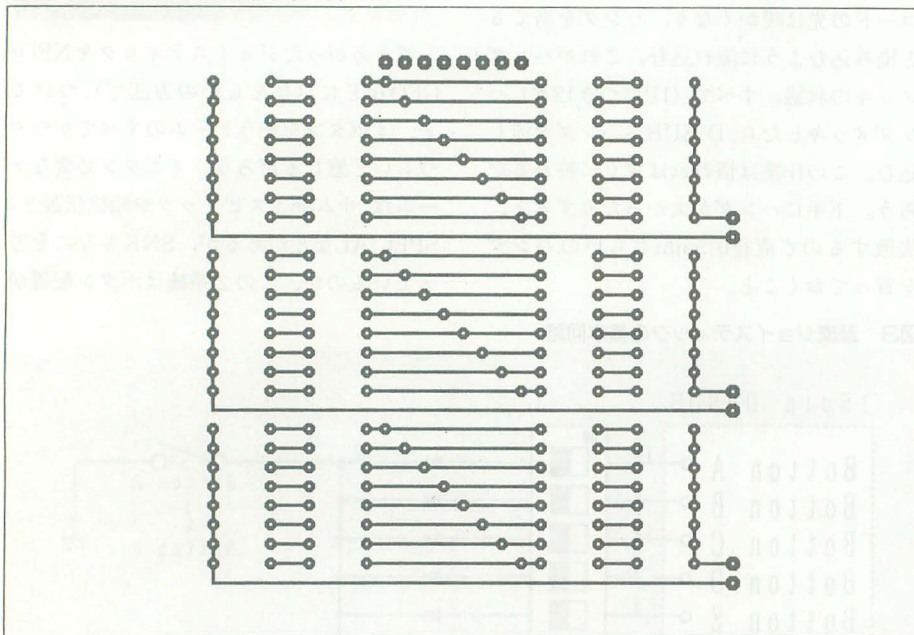
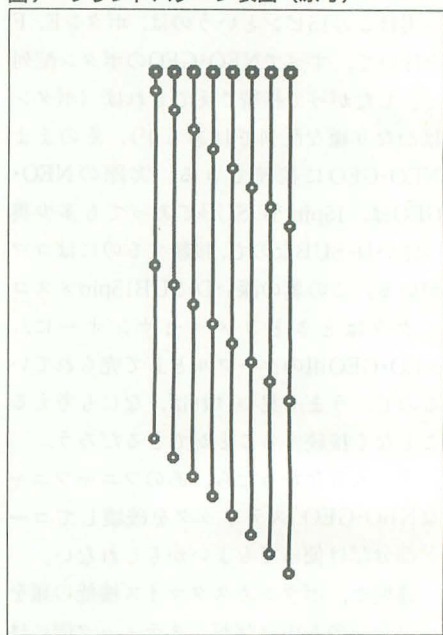



図7　プリントパターン表面（原寸）

いろいろなジョイスティック

うちにあるジョイスティックはすべて，この*NEO・GEO 15pin D-SUB*仕様だ。アダプタのせいでNEO・GEO純正ジョイスティックまで，X68000などにもつなげられる。実はケーブル＆小物ケースから滅多に出てこないが，15pin D-SUBのインタフェイスを持ったジョイパットまである。

本文で使用した6ボタンジョイスティックは確かに使いやすいが，ボタンが邪魔になるときもままある。それに，1Pと2Pが合体しているのでむやみに長いため，ひとりで膝の上にジョイスティックを置き，シューティングなどをやるにはちょっと不便になる。

そこで3ボタン1Pジョイスティックも別に持っているわけだ。このジョイスティックは3ボタンで，今回紹介したような，まったく素なジョイスティックではない。強烈なボタンカスタマイズ機能は外付けできるのだが，ジョイスティック内部にそれなりのカスタマイズ機能がある。

まずボタンA，Bだが，リバースができる。アタリ（MSX）準拠と呼ばれているスティックは，ボタンを右から数えるのが本当らしいのだが，一部左から数えるジョイスティックも発売されている。ソフトハウスによっては，左側から数えるスティックにあわせているところ，右側から数えるスティックにあわせているところの2種類がある。

たとえば，私は2ボタンの対戦ゲームは，左がパンチ，右がキックのほうがよい。強弱を決められるのなら，左が弱，右が強のほうがよい。

シューティングなんかは，左がショット，右がボムが望ましい。そういったわけで（本文のように大胆なボタンカスタマイズでもできるが），簡単に操作するためには，むしろ，1スイッチでA，Bリバース機能が必須になる。

そして3ボタン目だが，これはAボタンにもBボタンにもCボタンにもなる。拡張NEO・GEO 15pin D-SUBからATARI準拠9pin D-SUBに変換するコネクタの中に，CボタンはA＋Bの同時押しとする機能があるため，結果的にX68000やTOWNSなどで使うとき，ボタン配列は，A，B，A＋Bなどといった配置ができるのだ。

この設定で恩恵が得られるソフトは，たくさんある。思いつくところでは，餓狼伝説，チェルノブ，マッドストーカー，あすか120％などなど。マッドストーカーは，両ボタン押しで防御となるため，A，Bボタンを連射モードにできないでいた。A＋B同時押しが別にあれば，これを防御ボタンとすることができるので，存分に連射機能が使える。

で，以下のようにカスタマイズしたい。

A　小斬り

B　中斬り

C　大斬り

D　小蹴り

E　中蹴り

F　大蹴り

そのためにはまず，ボタンAの，8連ディップスイッチのボタンAと書かれた部分だけONにする。BはB，そしてCはAとBの部分をONにしてしまう。DはCだけ，EはDだけをONにし，最後にFはCとDをONにする。

こうすると，たちまち，ボタンが自分が好きなようにカスタマイズできることになるのだ。

回路の原理について説明しよう。図4を見ていただきたい。これはよくやる誤りの図である。単純に考えると，ボタンAとBの同時押しは，ボタンAとBから信号を奪ってくるだけでできる。確かに同時押しボタンだけの回路では間違いではない。しかし，図中の矢印のように，ボタンAを押しても，ボタンBまで押されたことになってしまっている。ボタンBを押しても同じことがいえる。

ボタンが押されるということは，最初のほうで説明したとおり，グランドレベルに落ちるということである。ということは，わずかな電流が結線を通じて，グランドに流れるにすぎない。この知識を応用して，間違った接続には，電流が流れないようにすればよい。それが図5である。ご覧のとおり，実によい位置にダイオードがいる。

これなら，矢印のような電流はカットされ，Aボタンを押してもBボタンまで押されるようになることはない。Bにしても同じ。さらにCボタンはきちんと同時押しボタンとして動作する。

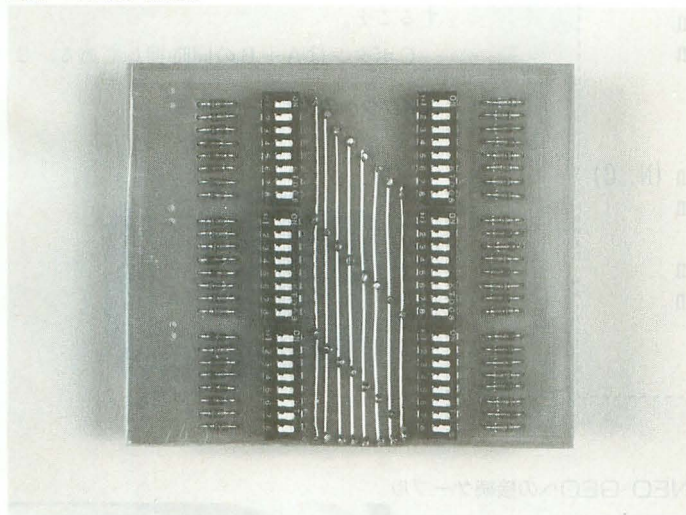
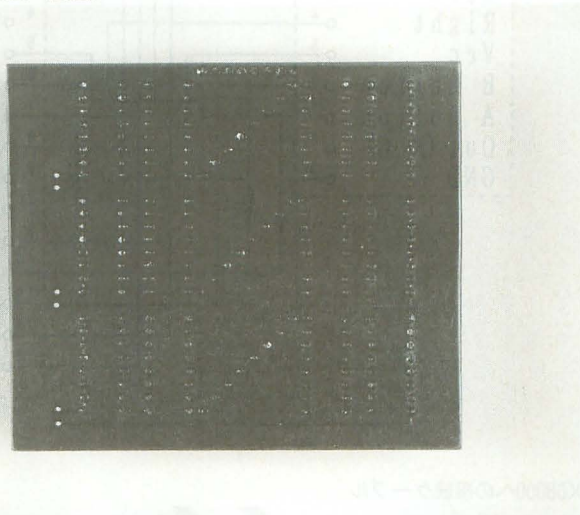
図5はただの2ボタン同時押しの回路であるが，そのまま図3の8ボタンスイッチつき同時押しの回路に発展することがわかるだろう。

図3の回路ではすべてのボタンが同時押しボタンだと考えてもらえばよい。ただ，途中でディップスイッチが入ることによって，電気的に回路が切断されるため，1ボタン押しだろうが，8ボタン同時押しだろうが関係なくできるわけである。

回路の説明が終わったところで，作成したプリント基板について説明しよう。図6,7がその裏面と表面である。実は両面基板のパターンを作ってみた。マトリクスになるので，ピン間に2,3本通せない限り，かなりうまくやらないとジャンパブリッジができてしまうのだ（うまい方法はあるけどさ）。

実装方法は写真を見ていただきたい。ただ，写真を見ればわかる通り，実は力技で片側の面をつなげている。図7のような，たったこれだけのプリントパターンのために，値段が2倍以上もするガラエポ両面基板を使うのはもったいなかったからだ(使った片面紙フェノールは380円，両面ガラスエポキシは880円)。

余談であるが，なぜセレクトボタンとスタートボタンを入れたかを書いておこう。ちなみに，これはセレクトボタンがAボタンになったりするのではないから，勘違いしないように。Aボタンがセレクトボタン

基板への実装（表面）

基板への実装（裏面）

スティックあれこれ

メッセサンオーに行くと，妙なジョイスティックのアングルがある。なかには8ボタンのジョイスティックアングルまであるので，試しに8ボタンのジョイスティックも作ってみた（これは私物）。

まず，アングルにぴったりあったスティックが必要だ。スティック，ボタンともに同店で販売しているが，探せばもっと安い店がある。もっとも，スティックは各社共通というわけではないので，同店で買ったほうがよいかもしれない（あそこに売っているスティックはあまり私好みではないが）。

ボタンのほうは同店の1/3ぐらいの値段で，売っているところもある。最近私が気にいっているのは強化ボタンだ。1個150円(どこの店だかは忘れた)。

こうして，すべてを揃えれば，1万円もかけずに「使いやすい」ジョイスティックができあがる。別に，HORIのジョイスティックとかが使いやすい人なら，それを改造してもいいんだけどね。ボタンは確かに問題はないけど，スティックは私が使うと，指が太いためか，アングルの間に指の肉を挟んでしまうんだな。

になったりするのだ。

さて，理由は簡単である。PCエンジンの世界では，ボタンが2つしかないため，3ボタンゲームをするときには，セレクトボタンで代用することが多いからだ。

いまでこそ6ボタンパッドが発売されたが，一昔前のPCエンジンスティックは，苦肉の策として，セレクトボタンをCボタンのように使う*4，スライドスイッチが別にあった。いまだに3ボタン使うようなゲームであっても，6ボタンパッドを使わずに，セレクトボタンで代用するものが多い。

やはり私のウチには当然のように，このスティックをPCエンジンにつなげるアダプタがあって，Cボタンがあるゲームが非常に不便しているというのも事実である（注：今回作成したスティックは編集部用であって私の私物ではない。私のスティックはAボタンをセレクトボタンにするという器用な真似はできない）。

とまぁ，余談がすぎたが，このようにして，ボタンカスタマイズ回路はできあがる。なお，この回路はスティックと本体との間につなげることをお忘れなく。

＊4　なぜ「ように」なのかは，ボタンのネーミングの違いである。PCエンジンは，右から順にI，II，IIIだからだ。この場合Cボタンのようにといっているのは，IIIボタンのことではあるが，場所的にはAボタンの位置にあることを忘れずに。

各機種への接続方法

当然ながら，まず最初にX68000への接続方法を書こう。結線図は図8だ。基本的に同じ機能を持つ接続を行っているが，多少工夫もある。まず，最近X680x0シリーズでもよく使われている，TOWNSパッド準拠の機能が埋め込まれている。このTOWNSパッド準拠というのは，割とよく考えられていて，9ピンという少ない配線数でボタンを増やすために，ひと工夫している。具体的には，絶対に同時に押されることがない，上下，左右がそれぞれ押されたときに，SELECTとRUNとして動作するようにしているのだ。もちろん，このSELECTとRUNがゲーム操作中に使われないことを前提に作られている。少し考えればおわかりのように，この2つのボタンはレバーと併用といった操作ができない。

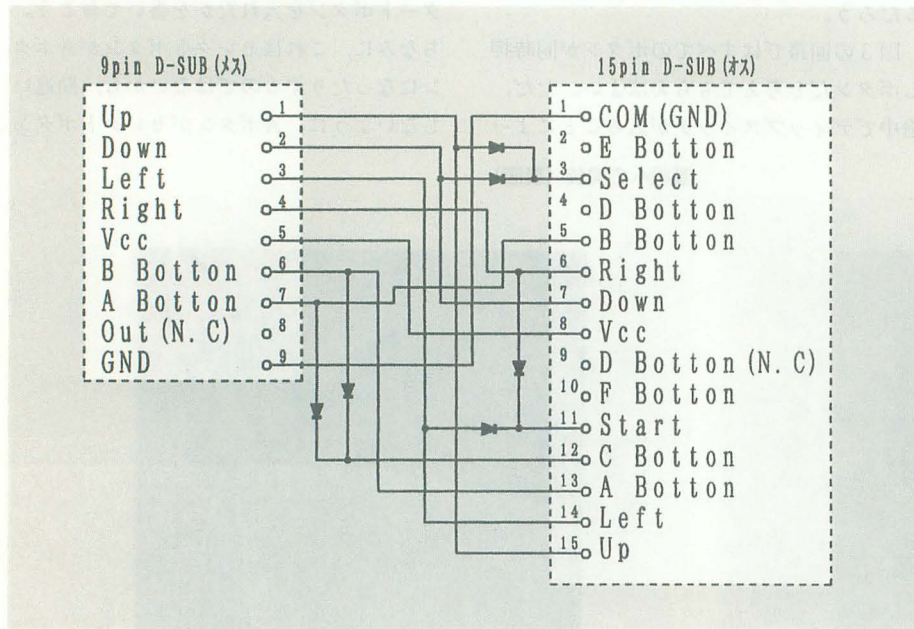
当然，ソフトだけの問題なので，X68000ほか，アタリ準拠と呼ばれているジョイスティックインタフェイスを持つ機種なら，対応できる。

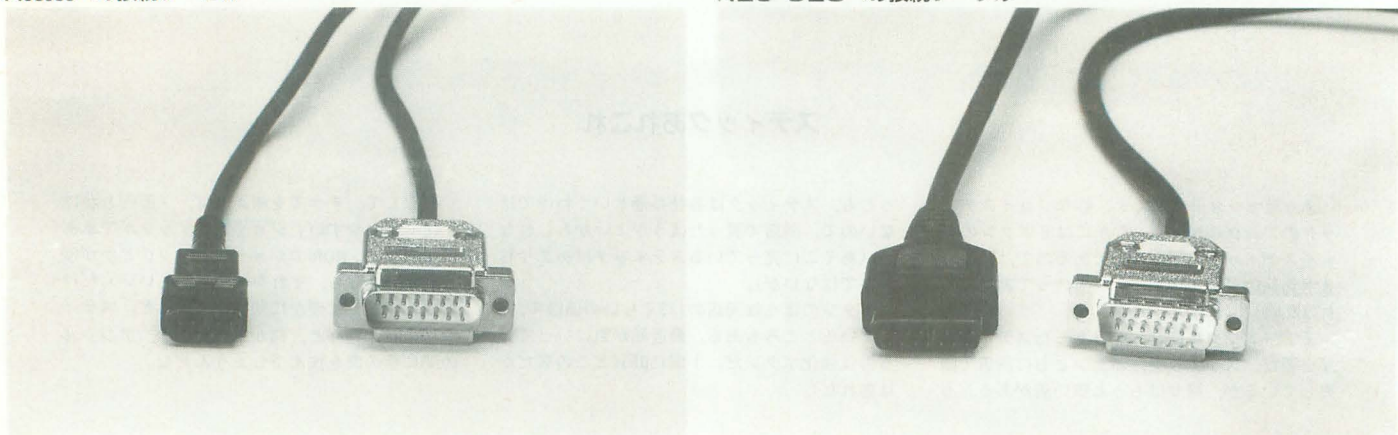
上下がSELECTで左右がRUNになっているが別に調べたわけではない。調べようと思えば，家にもTOWNSはあるので調べられるが，さして困らないのでこのようにした。

また，A，Bボタンがリバースしているが，この件についての理由はコラムを参照すること。

CボタンはA＋Bの同時押しである。こ

図8　X680x0とのインタフェイス

コネクタへのダイオード実装

X68000への接続ケーブル

NEO・GEOへの接続ケーブル

れもコラムを参照してほしい。

実際の結線で注意するのは，ダイオードの実装と9ピンD-SUBである。15ピンD-SUBメスの場合，普通のD-SUBでも支障がないことが多いので，一般に市販されているものを使う。しかし，マンハッタンシェイプのX68000の2P側を除き，ジョイスティック端子はくぼんでいるので，下手に汎用の9ピン D-SUBカバーをつけると，端子が接触するほど奥に入らない。ちょっと文章ではわからないがひとつ作ってみるとわかるだろう（もっとも作ってしまったら，そのケーブルはマンハッタンシェイプの2P側につなぐか，延長ケーブルの先にしか接続できないが）。

というわけでいちばんよいのは，PC-9801用マウス延長ケーブル，MSX用ジョイスティック延長ケーブル，メガドライブ用ジョイスティック延長ケーブルの，全結線のものを購入してきて，オス側をばらす方法である。9ピン D-SUBの延長ケーブルは多機種で同じなためか，割合安価である。下手に9pin D-SUBと9芯シールド線を数m買うよりも絶対に安上がりだ。

ダイオード周りの回路だが，このために基板を起こすのも面倒である。美しくまとめるには，15ピン D-SUBコネクタのカバーにうまく入るように実装するしかない。もはや立体配線の世界になってしまい，ピンセットなしでは作れないかもしれない。うまくまとめられたら，ホットボンドかなにかで固めて壊れないようにしてしまおう。

ちなみに写真でD-SUBの蓋が開いていないのは，実際にホットボンドでとめられてしまい，もはや開かないからでもある。

あぁ，おまぬけ。

代機種の接続方法についてもざっと述べておこう。

まず，NEO・GEO。このジョイスティック自身，拡張NEO・GEO配列なので，なんとか物理的に接触できればつながる。例のように怪しげな変換コネクタ基板を起こしてもよいが，こんなものはそうそうやすやすと部品が手に入るわけがない。もっとも現実的なのは，どこかでNEO・GEO用の延長ケーブルを探してくることか。

次に，メガドライブ。3ボタンと6ボタンがあるが，6ボタンは3ボタンを完全に包含するので，6ボタンで作ることをおすすめする。基本的にフラットパッケージのSEGAカスタムチップが1個入っている。信号をロジックアナライザで調べれば，おそらくTTLで等価回路が作れるだろう。ただ，こんなところで努力するよりも，パッドをばらして接続するほうが100倍は楽でお金もかからない。したがって，私はこちらに逃げた。写真がそれである。写真中のスイッチは，秘かにチェルノブアダプタを内蔵したと考えてもらって差し支えない。チェルノブアダプタについては，10月号の餓狼伝説SPECIALのレビューに結線図が掲載されているので，そちらを参考にしてもらいたい。

私の友人は，根性でフラットパッケージのICを抜き去り（これは簡単だが），このICをフィルム基板に張り付け，なんと15ピンD-SUBカバー内部に回路を内蔵してしまった。確かに異常なくらいに美しくまとまっており，実装技術には脱帽ものである。

同様にしてPCエンジンも作成できる。こちらは確か，HC157かHC153あたりのデータセレクタを2つぐらい使っていたと思う。切り換えスイッチはいるが，メガドライブと同じく2ボタン，3ボタン，6ボタンがセレクトできるので，うまく配線してほしい。

以上。サターンでも3DOでもPlaystationでも，パッドから抜き出して考えれば，簡単にお気にいりのスティックとつなげてしまうのだ。

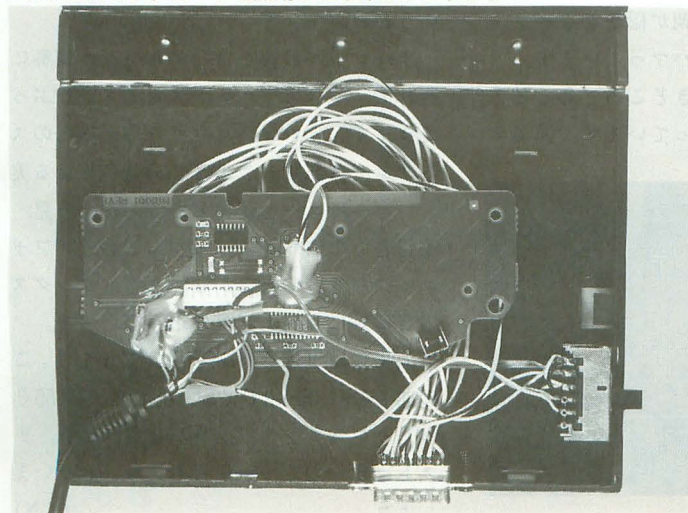
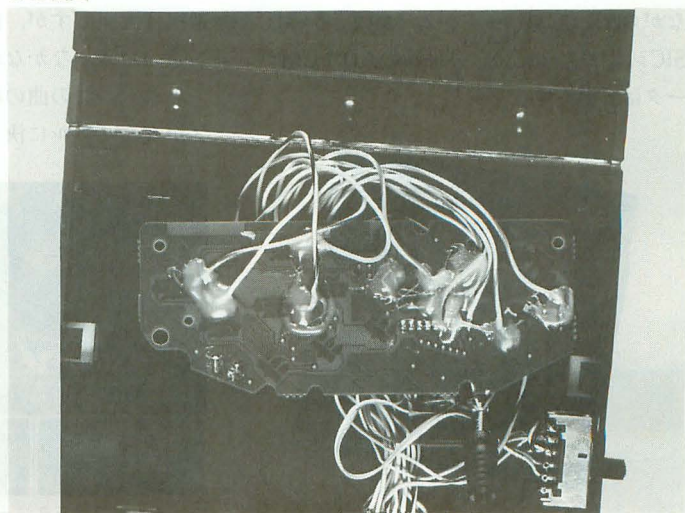
まとめ

今月は，まさにエゴイズムで原稿を書いたなぁ。独断と偏見って感じ。

最初から最後まで全力投球しないとなかなか終わらない連載仕事なので，毎月結構体力と時間を消耗する。前回など，最初に気合入れて，成功したからタラタラやるかと思ったら，結局全然終わらなかったしね。

こういった背景上，なかなか時間が取れないため，RS-MIDIなどの約束の回路はなかなか動作実験できない。頭の中には回路はあるんだけど。今回の連載においても，本当は外付けの連射回路など作りたかったところだ。

さて来月もなんとなく苦しくなりそうなことをやる予定だ。そろそろ始めないとヤバイか？　いったい，私はいつ，PCMボードの作成に時間を割くことができるのだろうか？　う〜ん。自殺行為だ。終わってる。合掌。

メガドラのコントローラを流用した6ボタンアダプタ

その裏面

Oh!X LIVE in '94

X68000・Z-MUSIC ver.2.0用

ダーク・スペース

Yabe Masatoshi
矢部　雅敏

X68000・Z-MUSIC ver.2.0用(SC-55対応)

ENDLESS RAIN

Chikira Kazuaki
千吉良　和明

X68000・Z-MUSIC ver.2.0用(SC-55対応)

©スクウェア「ファイナルファンタジーV」より

レナのテーマ

Osame Takafumi
納　享史

このところオリジナル曲の掲載が続いていますが，今月も1曲ご紹介します。楽器が演奏できなくても，パソコンなら好きなように曲を作ることができます。興味のある人はぜひ挑戦してみてくださいね。ほかの2曲もなかなかハイレベルな作品です。

シビレました，ホントに

まず1曲目は，内蔵音源のみを使用したオリジナル曲です。この曲，タイトルからも想像されるように「シューティングゲームのステージ1」といったイメージが強く出ています。イース系ARPGの草原のテーマにもなりそうです。カッコいいメロディとそれを盛り上げるバックパートの絡みが聴く者を熱くしてくれるでしょう。私はシビレました。いや，ほんと。

音源の本質的な使い方も素晴らしいのですが，それ以上に，自分の表現したいことを，制限された少ないパート数のなかで見事実現しているところがスゴイです。ぜひ入力して私同様シビレてください。

演奏には，PCM8.Xも不要。ただし，単音ながらAD PCMパートがあるので「Z-MUSICシステムver.2.0」に付属のAD PCMデータは必要です。

BLUE BLOOD

降り続ける雨

2曲目は，X JAPANのアルバム「BLUE BLOOD」から「ENDLESS RAIN」です。

作者の千吉良君はX68000EXPERTを半年前に買ったばかりのX68000初心者で，投稿も今回が初めてとのこと。しかし，イントロにZ-MUSICの特殊効果フェードインを用いたり，ギターの音色を局面に応じてNRPNで操作したりと，初心者とは思えない完成度です。リズムパートのアンビエンスエフェクトも完璧です。

ボーカル曲をコンピュータミュージックで実現するときのいちばんの悩みの種はメロディの音色です。これに関してやはり千吉良君も悩んだそうで，試行錯誤の結果，クワイア系の音色に落ち着いたとか。この選択は大胆ですが，曲調が穏やかなバラード系のため，なかなかハマっています。

それと，この曲の聴きどころであるピアノが実にリアルに決まっています。演奏データを覗いてみると，その苦労の形跡が至るところに……。この曲を聴くときには，一度はピアノをメインに聴いてみましょう。

演奏にはSC-55系のMIDI音源が必要です。内蔵音源は使用していません。SC-55mkIIやSC-88でもほとんどニュアンスの違いもなく聴けました。

ファイナルファンタジーV

アンサンブルなひととき

最後はゲームミュージックです。「ファイナルファンタジーV」から「レナのテーマ」です。ファミコンの「ドラクエ」「ファイナルファンタジー」シリーズなどの大作著名ゲームミュージックは投稿数が多いので，競争率は相当なものです。つまり逆をいえば掲載されたものはそれだけの競争にうち勝ったものなわけで，完成度はお墨つきというわけです。

オルゴールの寂しげなアルペジオ演奏に遅れて，泣きのフルートメロディがかぶってきます。これを盛り立てるのが後ろのストリングス部隊。この3つの音色による美しいハーモニーを受けとめるのが，一見，不似合いのE.ベース。しかし，このモノフォニックな低音の響きが荘厳なストリングスの音に妙に心地よく溶け込んでいます。

演奏にはSC-55系GS音源が必要です。この曲も，SC-55mkIIやSC-88でも，SC-55そっくりニュアンスで演奏がなされます。リストは短いので一気に打ち込んでしまいましょう。

(Z.N)

リスト1 ダーク・スペース

```
  1: .COMMENT ～ダーク・スペース～ Composed By Yabe.
  2:
  3: /  X68000+Z.MUSIC  Ver2.0
  4: /       Make 1994/01/24 ARG 07/28
  5:
  6: (i)
  7: (B0)
  8: (m1,3000)(a1,1)
  9: (m2,3000)(a2,2)
 10: (m3,3000)(a3,3)
 11: (m4,3000)(a4,4)
 12: (m5,3000)(a5,5)
 13: (m6,3000)(a6,6)
 14: (m7,3000)(a7,7)
 15: (m8,3000)(a8,8)
 16: (m9,3000)(a9,9)
 17:
 18: .ADPCM_BLOCK_DATA=DAS.ZPD
 19:
 20: (v2,0                /AT.BARSS
 21: /    AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 22:      61, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 23: /    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 24:      31, 10,  2,  3,  3, 40,  0,  4,  0,  0,  0
 25:      31,  3,  2,  8,  1,  9,  0,  2,  3,  0,  0
 26:      31,  3,  2,  7,  1,  2,  0,  1,  0,  0,  0
 27:      31,  3,  2,  7,  1,  2,  0,  2,  7,  0,  0)
 28:
 29: (v3,0                /A.PIANO
 30: /    AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 31:      44, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 32: /    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 33:      31,  5,  3,  6,  1, 35,  1,  2,  2,  0,  0
 34:      25, 10,  7,  7, 10,  5,  1,  2,  0,  0,  0
 35:      31,  5,  3,  6,  1, 32,  1,  1,  0,  0,  0
 36:      25, 10,  5,  7,  6,  2,  1,  2,  6,  0,  0)
 37:
 38: (v10,0               /BRASS
 39: /    AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 40:      58, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 41: /    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 42:      15,  9,  0,  5,  1, 25,  2,  2,  3,  0,  0
 43:      15,  9,  0,  5, 15, 31,  2,  2,  0,  0,  0
 44:      15,  0,  0,  5,  0, 25,  1,  2,  0,  0,  0
 45:      13,  3,  0,  8,  0,  0,  1,  2,  7,  0,  0)
 46:
 47: (v11,0               /
 48: /    AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 49:      61, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 50: /    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 51:      18, 10,  0,  6,  2, 25,  0,  1,  0,  0,  0
 52:      20,  5,  0,  8,  2,  0,  0,  1,  2,  0,  0
 53:      20,  5,  0,  8,  2,  0,  0,  1,  5,  0,  0
 54:      15,  5,  0,  8,  1,  0,  0,  1,  0,  0,  0)
 55:
 56: (v17,0               /MELODY
 57: /    AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 58:      61, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 59: /    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 60:      17,  4,  8,  0,  2, 26,  0,  2,  2,  0,  0
 61:      21,  8,  3,  7,  2,  9,  0,  1,  0,  0,  0
 62:      23,  8,  7,  7,  1, 10,  0,  2,  6,  0,  0
 63:      24,  7,  7,  7,  2,  8,  0,  4,  0,  0,  0)
 64:
 65: (v42,0               /BRASS2
 66: /    AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 67:      61, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 68: /    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 69:      25,  2,  0,  3,  0, 31,  0,  1,  1,  0,  0
 70:      26,  8,  5,  5,  3,  0,  2,  0,  5,  0,  0
 71:      24,  8,  5,  5,  3,  0,  2,  1,  4,  0,  1
 72:      26,  8,  5,  5,  3,  0,  2,  4,  4,  0,  0)
 73:
 74: (v137,0              /E.BASS
 75: /    AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 76:      56, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 77: /    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 78:      31, 18,  0,  6,  2, 36,  0, 10,  0,  0,  0
 79:      31, 14,  4,  6,  2, 45,  0,  0,  7,  0,  0
 80:      31, 10,  4,  6,  2, 18,  1,  0,  3,  0,  0
 81:      31, 10,  3,  6,  2,  0,  1,  0,  0,  0,  0)
 82:
 83: (v102,0              /S.BELL
 84: /    AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 85:      36, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 86: /    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 87:      26, 13,  2,  6,  1, 32,  0,  1,  3,  0,  0
 88:      23, 10,  2,  5,  1,  0,  0,  3,  7,  0,  0
 89:      16, 12,  1,  5,  1, 32,  0,  1,  7,  0,  0
 90:      16,  8,  1,  4,  1,  0,  0,  3,  3,  0,  0)
 91:
 92: (v20,0               /HI-HAT1
 93: /    AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 94:      59, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 95: /    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 96:      31,  0,  0,  7, 15, 15,  0, 14,  0,  1,  0
 97:      31, 10,  0,  9, 15, 27,  0,  8,  0,  0,  0
 98:      31, 25,  0, 10, 15, 15,  0,  6,  0,  0,  0
 99:      31, 18,  0,  8, 15,  0,  0,  0,  0,  0,  0)
100:
101: (v32,0               /HI-HAT2
102: /    AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
103:      44, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
104: /    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
105:      31,  5,  6,  0,  5,  0,  0, 15,  7,  1,  0
106:      31, 17,  4, 15, 15,  5,  0,  3,  0,  2,  0
107:      31, 20,  6,  3,  3,  0,  0,  1,  7,  2,  0
108:      31, 25,  6, 15, 15,  0,  2,  7,  0,  3,  0)
109:
110: (v100,0          /PSG+
111: /    AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
112:      60, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
113: /    AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
114:      25,  5,  4,  6,  3, 28,  1,  4,  3,  0,  0
115:      24,  6,  3,  7,  2,  2,  0,  6,  3,  0,  0
116:      25,  5,  4,  6,  3, 26,  1,  4,  7,  0,  0
117:      24,  6,  3,  7,  2,  3,  0,  6,  7,  0,  0)
118:
119: /----------------------------------------
120: /  T R K 1        M E L O
121: /----------------------------------------
122: (t1)r2[do] @17   18  o3 v14 ¯2  @m16 @s10 @h40
123: (t1)o4(c+32,d)&d2&d16.c16r16d16e-16r16c+16c+32&d8..c4>b-4g4
124: (t1)<c2&c8 c16r16d16e-16r16f+16 (F+16,G)&g8.f4e-4f8.&(f16,d)
125: (t1)O4(c+32,d)&D2&D16.  C16R16D16e-16r16c+16 d4c4>b-4g4
126: (t1)<c2&c8 c16r16q4d16q8e-16r16f+16 (f+16,g)&g8.f4e-4d4
127: (t1)_1c+32d32e-4..&e-8c16r16d16e-16r16f16f4d32e-8..d4e-4
128: (t1) d2&dc+32&d16.e-c& c2&ccde-
129: (t1) e32f4&f16.g4. q6ce-q8 d4.>b-4.g4
130: (t1)¯1a-4.a-16<c16e32&g8&g32f8.e-8 d16e-16d4.&d8>¯1b-8g8<d8
131: (t1) c32c+32d16q3d8q8e-c4>b-g<d q6c+64d8...q5e-8q8c4ce-g
132: (t1) f+32g8..a-f8.&q5e32e-32q8cde-
133: (t1) q7d32e-8..f+32f16.q8d4>a32&b-16.g8<d8
134: (t1)c32c+32d16q3d8q8e-c4>b-g<d q6d4q5e-q7c8.&(c16>g)<ce-g
135: (t1)f+32g8..a-f8.&e32e-32e-ca- g4.b-4&(b8,g)g4
136: (t1) _2 o4q6c4&c16r16q4c16d16 q7e-2&e-4 g4f4e-4
137: (t1) q6d4&d16r16q4d16e-16 q7c+32d8&d32c8.&c8& c1
138: (t1) o4q6c4&c16r16q4c16d16q7 e-2&e-4 f+32g8..b-4<e-4
139: (t1) q7d4.s2 @m40 @s8 @h40c4.&c4& c2.&(c4>g)¯2
140: (t1)[LOOP]
141: /----------------------------------------
142: /  T R K 2        M E L O   E C H O 1
143: /----------------------------------------
144: (t2)r8.R2[do] @17 18 o3 v11 @k-5 @m16 @s10 @h40 ¯1
145: (t2)o4(c+32,d)&d2&d16.c16r16d16e-16r16c+16c+32&d8..c4>b-4g4
146: (t2)<c2&c8 c16r16d16e-16r16f+16 (F+16,G)&g8.f4e-4f8.&(f16,d)
147: (t2)O4(c+32,d)&D2&D16.  C16R16D16e-16r16c+16 d4c4>b-4g4
148: (t2)<c2&c8 c16r16q4d16q8e-16r16f+16 (f+16,g)&g8.f4e-4d4
149: (t2)_1c+32d32e-4..&e-8c16r16d16e-16r16f16f4d32e-8..d4e-4
150: (t2) d2&dc+32&d16.e-c& c2&ccde-
151: (t2) e32f4&f16.g4. q6ce-q8 d4.>b-4.g4
152: (t2)¯1a-4.a-16<c16e32&g8&g32f8.e-8 d16e-16d4.&d8>¯1b-8g8<d8
153: (t2) c32c+32d16q3d8q8e-c4>b-g<d q6c+64d8...q5e-8q8c4ce-g
154: (t2) f+32g8..a-f8.&q5e32e-32q8cde-
155: (t2) q7d32e-8..f+32f16.q8d4>a32&b-16.g8<d8
156: (t2)c32c+32d16q3d8q8e-c4>b-g<dq6d4q5e-q7c8.&(c16>g)<ce-g
157: (t2)f+32g8..a-f8.&e32e-32e-ca- g4.b-4&(b8,g)g4
158: (t2) _2 o4q6c4&c16r16q4c16d16 q7e-2&e-4 g4f4e-4
159: (t2) q6d4&d16r16q4d16e-16 q7c+32d8&d32c8.&c8& c1
160: (t2) o4q6c4&c16r16q4c16d16q7 e-2&e-4 f+32g8..b-4<e-4
161: (t2) q7d4.s2 @m40 @s8 @h40c4.&c4& c2.&(c4>g)¯2
162: (t2)[LOOP]
163: /----------------------------------------
164: /  T R K 6        M E L O   E C H O 2
165: /----------------------------------------
166: (t6)r8 R2[do]@1718 o3 v11 ¯2 @k5 @m16 @s10 @h40 ¯1
167: (t6)o4(c+32,d)&d2&d16.c16r16d16e-16r16c+16c+32&d8..c4>b-4g4
168: (t6)<c2&c8 c16r16d16e-16r16f+16 (F+16,G)&g8.f4e-4f8.&(f16,d)
169: (t6)O4(c+32,d)&D2&D16.  C16R16D16e-16r16c+16 d4c4>b-4g4
170: (t6)<c2&c8 c16r16q4d16q8e-16r16f+16 (f+16,g)&g8.f4e-4d4
171: (t6)_1c+32d32e-4..&e-8c16r16d16e-16r16f16f4d32e-8..d4e-4
172: (t6) d2&dc+32&d16.e-c& c2&ccde-
173: (t6) e32f4&f16.g4. q6ce-q8 d4.>b-4.g4
174: (t6)¯1a-4.a-16<c16e32&g8&g32f8.e-8 d16e-16d4.&d8>¯1b-8g8<d8
175: (t6)c32c+32d16q3d8q8e-c4>b-g<dq6c+64d8...q5e-8q8c4ce-g
176: (t6) f+32g8..a-f8.&q5e32e-32q8cde-
177: (t6) q7d32e-8..f+32f16.q8d4>a32&b-16.g8<d8
178: (t6)c32c+32d16q3d8q8e-c4>b-g<d q6d4q5e-q7c8.&(c16>g)<ce-g
179: (t6)f+32g8..a-f8.&e32e-32e-ca- g4.b-4&(b8,g)g4
180: (t6) _2 o4q6c4&c16r16q4c16d16 q7e-2&e-4 g4f4e-4
181: (t6) q6d4&d16r16q4d16e-16 q7c+32d8&d32c8.&c8& c1
182: (t6) o4q6c4&c16r16q4c16d16q7 e-2&e-4 f+32g8..b-4<e-4
183: (t6) q7d4.s2 @m40 @s8 @h40c4.&c4& c2.&(c4>g)¯2
184: (t6)[LOOP]
185: /----------------------------------------
186: /  T R K 3        B A C K 1
187: /----------------------------------------
188: (t3)r2[do] @100 l16 o2 v9 @k-4 q8 p3 @m12 @s10 @h40
189: (t3) g2 b-4..<c& c4c4>b-4g4 <e-2 g2 f2 e-4d4
190: (t3) e-2 f4..d&  d2c2 d2  e-2 f2 e-4d4
191: (t3)@10 @k6 ¯3 p1o2a-2<c2 e-2d2 c2e-2   g2f2
192: (t3)@17 @k3 o4 v12 c4.c4.&c4 >b-4.f4.&f4
193: (t3)e-2<c2>g2 @102_8o3l16q4cp1gp3b-<p2cp3gp1b-<p2cp3g
194: (t3)@2 q8 o4 v9
195: (t3)|:c1>b-2<c2 >a-1 b-4.b-4.&b-4:|
196: (t3)@3 o4 v11 q8 f4.f4.f8.<e-&
197: (t3)e-4>¯1f4¯2f4_1f4_2 <c4.c4.c4
198: (t3)r8c8c4c8.c8.r8 o4f4.f4.f8.<e-&
199: (t3)e-4>¯1f4¯2f4_1f4_2@10o2a-4.<c4.e-4
200: (t3)g4.f4 @42o5v5¯1|:12g32¯1:|[LOOP]
201: /----------------------------------------
202: /  T R K 4        B A C K 2
203: /----------------------------------------
204: (t4)r2[do] @100 l16 o2 v9 @k4 q8 p3 @m12 @s10 @h40
205: (t4)g2 b-4..<c& c4c4>b-4g4 <e-2 g2 f2 e-4d4
206: (t4)e-2 f4..d&  d2c2 d2  e-2 f2 e-4d4
207: (t4)r32 _6 @k-6 @10 ¯5 p2o2a-2<c2 e-2d2
208: (t4)c2e-2g2f2 @17 @k-3 o4 v12 _5 c4.c4.&c4>b-4.f4.&f4
209: (t4)e-2<c2>g2@102q4_5o3l16q4 cp1gp3b-<p2cp3gp1b-<p3cp2g32
210: (t4)@2q8 o4 v9 p1
211: (t4)|:e-4@11p3v11o4q5g&_9g¯9q2fq4g&_9g¯9rrrr4 q8@2o4v9p1d2e-2
212: (t4) @10o2v12 |a-4<c4e-4a-4 g4.f4.>b-4@2o4v9p1 :|
213: (t4) o2a-4b-4<c4e-4 d4.f4.>b-4 @3o4v11 q8
214: (t4)a-4.a-4.r16a-8. a-4a-4a-4a-4
215: (t4)<e-4.e-4.e-4 r8e-8f4 d8.e-8.&e-8>
216: (t4)o4a-4.a-4.r16 a-8. a-4a-4a-4a-4
```

```
217: (t4)@10 o3 c4.e-4.a-4
218: (t4)o4 d4.c4 @42o5v5~1|:12b-32~1:|[LOOP]
219: /-------------------------------------------
220: /  T R K 5         S U B
221: /-------------------------------------------
222: (t5)r2[do] @102 l16 o3 q2 v5 p3 @m12 @s10 @h40
223: (t5)|:8o4ce-g<c>~1:||:4o3a-<ce-a->:||:o3f32&_6f32~6a-<cf>:|
224: (t5)|:o3g32&_6g32~6b-<dg>:|
225: (t5)|:8o4ce-g<c>:||:4o3a-<ce-a->:||:o3f32&_6f32~6a-<cf>:|
226: (t5)|:o3g32&_6g32~6b-<dg>:||:8R1:| @2 o4 v9 p2
227: (t5)|:g4@11p3v11o5q5c&_9c~9>q2b-<q4c&_9c~9rrr
228: (t5)r4 @2q8o4v9p2r4 f16g16f16g16 f16g16e-4.
229: (t5)@2o4v9p2 E-1f4.f4.f4@2o4v9:|@3 o5 v11 q8
230: (t5)c4.e-4. r8c8 c4~1e-4~2e-4_1e-4 _2
231: (t5)g4.g4.g4 r8g8g4 f8.g8.&g8
232: (t5)o4c4.e-4.r8c8 c4~1e-4~2e-4_1e-4_2
233: (t5)@10 o3 e-4.a-4.<c4 o4 g4.f4r4.
234: (t5)[loop]
235: /-------------------------------------------
236: /  T R K 7         E . B A S S
237: /-------------------------------------------
238: (t7)r2[do] @137 o2 l16 v13
239: (t7)|:o2<c8>c8 <c8>c<c& cq2gq8>c8<c8>c8
240: (t7)  o2<c8>c8 <c8>c<c& cq2gq8>c8<c8>c8
241: (t7)  <a-8>a-8 <a-8>a-<a-& a-q2a-q8>a-8<a-8>a-8
242: (t7)  <f8>f8 <f8>f<g& gq2gq8>g8<g8>g8  :|
243: (t7)o2<a-8>a-8 <a-8>a-<a-& a-q2a-q8>a-8<a-8>a-8
244: (t7)o2<b-8>b-8 <b-8>b-<b-& b-q2b-q8>b-8<b-8>b-8
245: (t7)o2<c8>g8 <c8>c<c& cq2g>q8c8<c8>c8
246: (t7)o2<e-8>e-8 <e-8>e-<e-& e-q2e-q8>b-8<e-8>e-8
247: (t7)  o2<f8>f8  <f8>f<f& fq2fq8>f8<f8b-8
248: (t7) o2<c8>c8 <c&d>c<c& cq2gq8>c8g8<c8>
249: (t7) o2<a-8>a-8 <a-8>a-<a-& a-q2a-q8>a-8 <a-8>a-8
250: (t7) <g8>g8<b-8>b-8<<d8>d8>><b-8>b-8
251: (t7) |:o2<c8>c8 <c8>c<c& cq2gq8>c8<c8>c8:|
252: (t7) o2<a-8>a-8 <a-8>a-<a-& a-q2a-q8>a-8<a-8>a-8
253: (t7) o2<b-8>b-8 <b-8>b-<b-& b-q2b-q8>b-8<b-8>b-8
254: (t7) |:o2<c8>c8 <c8>c<c& cq2gq8>c8<c8>c8:|
255: (t7) o2<a-8>a-8 <a-8>a-<a-& a-q2a-q8>a-8<a-8>a-8
256: (t7) o2<g8>g8 <g8>g<g& gq2gq8>g8<g8>g8
257: (t7) o2   ffrf <f8>q2fq8f& f<cf>f <f8>ff&
258: (t7)      ffrf <f8>ff& f<cf>f <f8>ff
259: (t7)     <ccrc <c8>q2cq8c& cg<c>c <c8>c<c&
260: (t7)      c>crc <c8>cc& ce-gc <c8>cc
261: (t7)      ffrf <f8>ff& f<cf>f <c8>ff&
262: (t7)      ffrf <f8>ff& f<cf>f <f8>ff
263: (t7)      a-a-ra- <a-8>a-a-& a-<e-a->a- <a-8>a-b-&
264: (t7)o2b-b-rb- <b-8>b-b- o2<g>go2<b->b- o3<d>do2<b->b-
265: (t7)[loop]
266: /-------------------------------------------
267: /  T R K 8         H I - H A T
268: /-------------------------------------------
269: (t8)@20o5v9|:8c16~2:| [do]L16 O3 @K-3
270: (t8)@20o5v10|:16q8cq2ccc  q8c8q2cc:|
271: (t8)@20 o4 l16 v11
272: (t8)|:8q8p3c8c8 p1c8p2c16c16 p3c8c8p1c8p2c16c16:|
273: (t8)@32 v11 o5  l16|:16q8cq2ccc q8cq2ccq8|c&:|c
274: (t8)|:8r1:| [LOOP]
275: /-------------------------------------------
276: /  T R K 9         D R U M S
277: /-------------------------------------------
278: (t9)t149 @f4 l16 o1 c<<P1f32f32P3eP2d32d32 P3>C8>c<C> [do]
279: (t9)|:4o4c+8ee  o1<C8>co4e o1cro4ee o1<C8>co4c
280: (t9) o1cro4ee o1<C8>co4e |o1cro4ee o1<C8>co4c :|
281: (t9) o2cro3ba o4p1c+8p2c+8&c+8o1p3c8 <C8>c8 o4eeo1c8<C8>c8
282: (t9)|:3 o1c8o4ee o1<C8>c8 o4eeo1c8<C8>co4e:|
283: (t9)    o4c+8o4ee o1<C8>c8 o4eeo1c8<C8>c<C>
284: (t9)    o1c8o4ee  o1<C8>c8 o4eeo1c8<C8>c<C>
285: (t9)    o4c+8o4ee o1<C8>c8 o4eeo1c8<C8>co4e
286: (t9)    o1c8o4ee  o1<C8>c<c rc32c32>cr<<b32a32r>>c<C>
287: (t9) o4c+8o4ee o1<C8>c8 o4eeo1c8<C8>co4e
288: (t9)|:6 o1c8o4ee o1<C8>c8    o4eeo1c8<C8>co4e:|
289: (t9)    o1c8o4ee o1<C8>CO4e  o2c8>c<C8>c<CC>
290: (t9)    o4c+8o4ee o1<C8>co4e ro1ccr <C8>co4e
291: (t9)|:2o1co4eee o1<C8>co4e   ro1ccr <C8>co4e :|
292: (t9)    o1co4eee o1<C8>rc    ro1ccr <C8o5p1q4cp2cp3q8
293: (t9)    o4c+8o4ee o1<C8>co4e ro1ccr <C8>co4e
294: (t9)|:2o1co4eee o1<C8>co4e   ro1ccr <C8>co4e :|
295: (t9)   o1co4eee o1<C8>c<C8C>cr<C32C16.>c<C>
296: (t9)[loop]
297:
298: (p)
```

リスト2　ダーク・スペースの音色コンフィグファイル

```
.O1C =FCK.PCM,,V80
.O2C =SHPS.PCM,,V80
.O3D =TOM7.PCM,,V83
.O3E =TOM6.PCM,,V83
.O3F =TOM5.PCM,,V83
.O3A =POWT3.PCM,,V80
.O3B =POWT4.PCM,,V80
.o4C#=CRSH1.PCM,,V80,MO1C
.O4E =HH2.PCM,,V60
.O5C =CLP808.PCM,,V80
```

リスト3　ダーク・スペースのカウンタ表示

```
1:00000060 00001800   2:00000084 00001800   3:00000060 00001800   4:00000060 00001800
5:00000060 00001800   6:00000078 00001800   7:00000060 00001800   8:00000060 00001800
9:00000060 00001800
```

リスト4　ENDLESS RAIN

日本音楽著作権協会（出）許諾第9471720-401号

```
1: /---------------------------------------------------------------
2: .comment  Endless Rain composed by YOSHIKI  programed by sotuo
3: (i)
4: (b1)
5:
6: (m1,3000)(amidi1,1)
7: (m2,3000)(amidi2,2)
8: (m3,7000)(amidi3,3)
9: (m4,6000)(amidi3,4)
10: (m5,3000)(amidi4,5)
11: (m6,3000)(amidi10,6)
12: (m7,3000)(amidi10,7)
13: (m8,3000)(amidi10,8)
14: (m9,3000)(amidi5,9)
15: (m10,3000)(amidi6,10)
16: (m11,3000)(amidi7,11)
17: (m12,3000)(amidi12,12)
18: (m13,3000)(amidi13,13)
19: /---------------------------------------------------------------
20: .roland_exclusive 16,66{$40,00,$7F,00}
21: .sc55_chorus{2,1,60}
22: .sc55_reverb{3,3,0,90,80}
23: .sc55_v_reserve $10{3,5,5,1,1,1,1,0,0,5,0,1,1,0,0,0}
24: /---------------------------------------------------------------
25: (o135)
26:
27: (t1) @i$41,$10,$42 @e70,35
28: (t2) @i$41,$10,$42 @e60,30
29: (t3) @i$41,$10,$42 @e60,10
30: (t4) @i$41,$10,$42 @e70,10
31: (t5) @i$41,$10,$42 @e60,30
32: (t6) @i$41,$10,$42 @e100,30
33: (t7) @i$41,$10,$42 @e100,30
34: (t8) @i$41,$10,$42 @e100,30
35: (t9) @i$41,$10,$42 @e30,30
36: (t10) @i$41,$10,$42 @e90,15
37: (t11) @i$41,$10,$42 @e90,15
38: (t12) @i$41,$10,$42 @e90,15
39: (t13) @i$41,$10,$42 @e90,15
40: (t1)@m @q0 @g12@b0@p64@51@v 04@u60 o4@k0  /guitar1,strings
41: (t2)@m @q0 @g12@b0@p64@51@v 04@u60 o3@k0  /guitar2,strings
42: (t3)@m @q0 @g12@b0@p64@ 2@v110@u75 o5@k0  /piano
43: (t4)@m @q0 @g12@b0@p64@ 2@v110@u95 o3@k0  /piano
44: (t5)@m @q0 @g12@b0@p64@36@v110@u85 o4@k0  /base
45: (t6)@m @q0 @g12@b0@p64@ 1@v110@u90 o3@k0  /synbal
46: (t7)@m @q0 @g12@b0@p64@ 1@v110@u110 o3@k0  /snere drum
47: (t8)@m @q0 @g12@b0@p64@ 1@v110@u90 o3@k0  /bass drum
48: (t9)@m @q0 @g12@b0@p64@44@v110@u80 o3@k0  /base
49: (t10)@m @q0 @g12@b0@p78@30@v105@u90 o4@k0  /solo
50: (t11)@m @q0 @g12@b0@p78@43@v105@u90 o4@k0  /solo
51: (t12)@m @q0 @g12@b0@p64@53@v105@u90 o3@k0  /vocal
52: (t13)@m @q0 @g12@b0@p64@53@v 90@u90 o4@k0  /chorus
53: /---------- guitar1 ----------------------------------------
54: (t1)
55: y$63,$01 y$62,$63 l16|:48~2'deg'&:|l1'deg' t7712
56: y$06,$4D u+10'eg'u-10 y$06,$42 'dg''cg''dg'
57: y$06,$4A u+12'ceg'u-6 y$06,$42 '>b<dg'u-6'>a<cg''>b<dg'
58: y$06,$50 u+20'c>af1''c>af'u-10 y$06,$42
59: '>bge'y$06,$51 u+12'cfg' y$06,$42 u-12'ceg'u-10'cfg''ceg'
60: l1|:3rrrr:| r|:3rrrr:|
61: /[E]
62: o4l2u65g1a1a<cff8e8d4e1e1fgab l1|:3rrrr:|
63: /[H]
64: l2edcdfgadedceagab
65: /[I]
66: t77@v100o6c2&l8c>a-b-<cd2.>b-4<c2&cc16e-16ggf2f4f4
67: /[J]
68: l2e-dc>b-a-b-<cde-fgdce-f>b-
69: /[K]
70: <cd>b-b<cde-d cd>b-ba-b- <c1>b1
71: /[L]
72: |:6r1:| y$06,$49 u100@120o3c1u65r1 y$06,$42
73: /[M]
74: @51|:5r1:|
75: /[N]
76: o5edcefgag4f4 eda1agab eda1fgab8<c8d8>b8
77: <c>b<c1fgab e1e1fgfg4d4 ¥21 c>b<c1fgab
78: /---------- guitar2 ----------------------------------------
79: (t2)
80: l16|:48~2c&:|l1c |:4c&:| >fg<c&c
81: l1|:3rrrr:|r
82: /[C]
83: o3@v90u85 l8 @d1 @27ceg<ded>ge@d0@d1
84: cdgb<g>bgd@d0@d1 >a<ea<cec>ae@d0@d1
85: fa<cf@d0@d1 gd>bg@d0@d1
86:
87: ceg<ded>ge@d0@d1 cdgb<g>bgd@d0@d1 >a<ea<c@d0@d1
88: >fa<cf@d0@d1 >g<cdg@d0@d1 d>bgd@d0@d1
89: /[D]
90: o3fa<cf@d0@d1 >gb<dg@d0@d1 >cg<ce@d0@d1 >eg+b<e@d0@d1
91: >fa<cf@d0@d1 >gb<dg@d0@d1 >cg<cf@d0@d1 ec>ge@d0@d1
92: /[E]
93: l8@27@v90u85 |:o3ceg<c@d0@d1 >>b<dgb@d0@d1
94: >a<ea<c@d0@d1 >>g<ea<c@d0@d1 >fg<cf@d0@d1 >eg<ce@d0@d1
```

▶10月号の99ページを見て，「お，僕の誕生日って(で)さんと同じだったんだぁ。なんとなく嬉しいな」と思ったのも束の間，今年で18歳になりました。もう悪いことはできません。
太田　志輝(18)北海道

```
 95: >dg<cf@d0@d1 >>g<dgb@d0@d1:|
 96: /[F]
 97: o3ceg<ded>ge@d0@d1 cdgb<g>bgd@d0@d1
 98: >a<ea<cec>ae@d0@d1 fa<cf@d0@d1 gd>bg@d0@d1
 99:
100: ceg<ded>ge@d0@d1 cdgb<g>bgd@d0@d1
101: >a<ea<c@d0@d1 >fa<cf@d0@d1 >g<cdgd>bgd@d0@d1
102: /[G]
103: fa<cf>gb<dg@d0@d1 >cg<ce>eg+b<e@d0@d1
104: >fa<cf>gb<dg@d0@d1 >cg<cfec>ge@d0@d1
105: /[H]
106: o3l8ceg<c@d0@d1 >>b<dgb@d0@d1 >a<ea<c@d0@d1 >>g<ea<c@d0@d1
107: >fg<cf@d0@d1 >eg<ce@d0@d1 >dg<cf@d0@d1 >>g<dgb@d0@d1
108:
109: ceg<c@d0@d1 >>b<dgb@d0@d1 >a<ea<c@d0@d1 >>g<ea<c@d0@d1
110: >fg<cf@d0@d1 >eg<ce@d0@d1 >dg<cf@d0@d1 >>g<dgb@d0
111: /[I]
112: o2l8@v90@27u100|:|:8'a-<e-a-<ce-a-':||:8'a-<fa-<d':|:|
113: /[J]
114: o4@30@p50@e90,30 u125l8r1r1r1gf<c>b-<c>b-<b-a-
115: l16g4.a-gf8.>b-<fr32ga-.gfe-4fdd4(a-b-32)&a-16.a-g
116: c4.>(a-<c)&>a-b-4a-@q1b-<cd@q0e-cde-fde-fge-fgr4
117: /[K]
118: o4r4e-4d8e-8f8g8 e-4>b-8g8g8b8<d8f8
119: e-dc4.>b-8<c8d8f8 e-4<g8l24fgfe-fe-de-dcdc>b<c>b
120: l16a-4<(de-)&d8.d8e-8frgf e-4d8e-8>gb<dfe-dcd
121: cde-fge-fga-fga-b-ga-b- g2.u-20(a-b-32)&u-10a-8&(a-a-)&a-32g1
122: / [L]
123: @e60,30 |:8r1:|
124: /[M]
125: o3l8@27@p64u85 @d1
126: fa<cf@d0@d1 >gb<dg@d0@d1 >cg<ce@d0@d1 >eg+b<e@d0@d1
127: >fa<cf@d0@d1 >gb<dg@d0@d1 >cg<cf@d0@d1 ec>ge@d0@d1
128: cg<cf@d0@d1 ec>ge@d0@d1
129: /[N]
130: |:5o3ceg<c@d0@d1 >>b<dgb@d0@d1 >a<ea<c@d0@d1 >>g<ea<c@d0@d1
131: >fg<cf@d0@d1 >eg<ce@d0@d1 >dg<cf@d0@d1 >>g<dgb@d0@d1:| ¥21
132: o3ceg<c@d0@d1 >>b<dgb@d0@d1 >a<ea<c@d0@d1 >>g<ea<c@d0@d1
133: >fg<cf@d0@d1 >eg<ce@d0@d1 >dg<cf@d0@d1 >>g<dgb@d0@d1
134: /----------- piano(r.h.) ----------------------------------------
135: (t3)
136: l1rrrr @d1
137: l8|:|:u+8'ce'u-8>g<:|@d0@d1 |:u+5'>b<d'u-5>g<:|@d0@d1
138: |:u+9'ce'u-9>a<:|@d0@d1 |:u+9'>b<d'u-9>g<:|@d0@d1 :|
139: >u+28'fa'u-28cu+18'a<c'u-18f
140: u+10'<cf'u-10a<u+15'af'u-15c@d0@d1
141: >u+15'fa'u-15c'a<c'au+5'gb<d4'u-5 l64gb<d32l8e16f@d0@d1
142: |:|:u+14'cf'u-14>g<:|@d0@d1 |:u+14'ce'u-14>g<:|@d0@d1 :|
143: /[A]
144: >u89'ce'>u-17g<cdu+12'g<c'u-9e'eg'c
145: u+8'd>b'u-13>gb<du+15'gb'u-10d'd>b'u+10'f>g+'
146: 'ce'u-13>a<ceu+11'a<c'u-8e'ea'u-10cu+10
147: 'fc'>a<'df'>a'b<d'g'b<d'g
148:
149: u85<'ce'u-13>g<ceu+15'g<c'u-12eu+10'eg'u-10c
150: u+10'd>b'u-10>gb<du+9'gb'u-9d>b<u+10'f>g+'
151: 'ce'u-14>a<ceu+19'fa'u-15cu+10'fa'u-10c
152: u+8'cd'>u-8g<u+8'cd'>u-8gu+5'b<e'u-5g'b<d'g
153:
154: <|:u+4'cf'>u-4a<:||:u+5'dg'u-5>b<:|
155: |:u+6'eg'u-6c:||:u+7'eg+'>u-7b<:|
156: |:u+8'fa'u-8c:||:u+12'dg'u-12>b<:|
157: |:|:u+11'cf'u-11>g<:||:u+11'ce'u-11>g<:|:|
158: / [C]
159: u102'ce'>u-17g<cdu+12'g<c'u-9e'eg'c
160: u+8'd>b'u-13>gb<du+15'gb'u-10d'd>b'u+10'f>g+'
161: 'ce'u-13>a<ceu+11'a<c'u-8e'ea'u-10cu+10
162: 'fc'>a<'df'>a'b<d'g'b<d'g
163:
164: u90<'ce'u-13>g<ceu+15'g<c'u-12eu+10'eg'u-10c
165: u+10'd>b'u-10>gb<du+9'gb'u-9d'd>b'u+10'f>g+'
166: 'ce'u-14>a<ceu+19'fa'u-15cu+10'fa'u-10c
167: u+8'cd'>u-8g<u+8'cd'>u-8gu+5'b<e'u-5g'b<d'g
168:
169: <|:u+4'cf'>u-4a<:||:u+5'dg'u-5>b<:|
170: |:u+6'eg'u-6c:||:u+7'eg+'>u-7b<:|
171: |:u+8'fa'u-8c:||:u+12'dg'u-12>b<:|
172: |:u+10'cf'u-10>g<:|u+13'eg'u-13c16u+9'eg16'u+9'eg'u-18c
173: / [E]
174: o4l8|:u+10'ce'u-10>g<:||:u+10'd>b'u-10>g<:|
175: |:4u+10'ce'u-10>a<:|
176: |:u+12'cf'u-12>a<:||:u+9'eg'u-9c:|
177: |:u+9'fa'u-9c:||:u+8'cd'u-8>g<:|
178:
179: |:u+10'ce'u-10>g<:||:u+10'd>b'u-10>g<:|
180: |:4u+10'ce'u-10>a<:|
181: |:u+12'cf'u-12>a<:||:u+9'eg'u-9c:|
182: |:u+9'fa'u-9d:||:u+12'df'u-12>b<:|
183: / [F]
184: o4l8u104'ce'u-9>g<c'ed.'l16cegu+9cu-9eg<c
185: l8u+12>'gb<d'u-12dgbu+15l64gb<d16.l8u-15>du+17'b<dg'u-5'b<df'
186: 'a<ce'u-12eabl16<u+10c>u-10bab<c>ba8
187: f8'c>a8.'u+17'fa<ce8.''gb<d8.''dgb8''eg<c8''dgb'
188:
189: 'eg<c2'&'eg<c'u-19eg<cu+9eu-9>g<ceu+11
190: 'd>gb2'l64dgb8&b32l64eg<c32&c8>'gb<d8'
191: l16u+7'ea<c4'&'ea<c''<ce''a<c''ea'
192: l24'a<c'u-5'ea''ec'u+9'ea'u-9'ec'u+6'c>a'
193: u+3'ec'u-9'c>a''>ae'u+8'c>a'u-8>'ae''ec'u+6
194: l16'd>b4'&'d>b'u-6'b<d'u-10g8u+10'b<e8'u-10g8'b<d8'g8
195: /[G]
196: u104'a<cf4'u-9cfa8u+9'a<cf8.''b<dg8''gb<d8''g<ce'&
197: 'g<ce8'u-9g8'<ce8'g8u+9'b<e8'u-9g+8'b<d8'g+8
198: u+9'fa<c4'u-9cfafu+9'a<cf8.''b<dg8'<'dgb8.'
199: 'fg<c4''cf8'u-9>g8'<ce8'g8<c8g8
200: /[H]
201: l8o4u90|:u+10'ce'u-10>g<:||:u+10'd>b'u-10>g<:|
202: |:4u+10'ce'u-10>a<:||:u+12'cf'u-12>a<:||:u+9'eg'u-9c:|
203: |:u+9'fa'u-9c:||:u+8'cd'u-8>g<:|
204:
205: |:u+10'ce'u-10>g<:||:u+10'd>b'u-10>g<:|
206: u+10'ce'u-10>a<'ce8.''ea<c16'u+9'ea<c'u-9c'ea'c
207: |:u+9'fa'u-9c:||:u+9'eg'u-9c:||:u+9'fa'u-9d:||:u+9'dg'u-9>b<:|
208: /[I]
209: o4l8u112|:|:'ce-a-4':|'e-a-<c4''a-<ce-4'
210: 'dfb-4''dfb-.'u-12'fb-<d16'u+12
211: 'fb-<d.'u-12'b-<df16'u+12'b-<df4':|u-12
212: /[J]
213: u90|:u+9'e-g'u-9>b-<:||:u+9'df'u-9>b-<:|
214: |:u+9'ce-'u-9>g<:||:u+9'd>b-'u-9>f<:|
215: |:u+8'ce-'u-8>a-<:||:u+9'e->b-'u-9>g<:|
216: |:u+9'cf'u-9>a-<:||:u+9'df'u-9>b-<:|
217:
218: |:u+9'e-g'u-9>b-<:||:u+9'df'u-9>b-<:|
219: |:u+9'ce-'u-9>g<:||:u+9'd>b-'u-9>f<:|
220: |:u+8'ce-'u-8>a-<:||:u+9'e->b-'u-9>g<:|
221: |:u+9'cf'u-9>a-<:|u+9'ce-'u-9>a-<u+4'df'u-4>b-<
222: /[K]
223: |:3|:u+9'ce-'u-9>a-<:||:u+9'df'u-9>b-<:|
224: |:u+9'e-g'u-9>b-<:||:u+9'dg'u-9>b<:|:|
225: |:u+9'a-<c'u-9e-:||:u+9'b-<d'u-9f:|
226: r>g<cdu+9gu-9<cdgd>gb<du+12'gb2'
227: /[L]
228: >u99'ce'>u-14g<ceu+9<c>u-9g'g<c'e
229: u+8'd>b'u-8>gb<d'gb'd'd>b'u+10'f>g+'
230: u+8'ce'u-18>a<ceu+8'a<c'u-8e'ea'u-10cu+22
231: 'fc'u-14>a<u+9'df'u-9>au+7'<cd'u-7g'b<d'g
232: u+9<'ce'u-10>g<ceu+13'<ce'u-13g'g<c'e
233: u+9'dg'u-9>b<dgu+12'b<d'u-12gdu+9'a-f>b'
234:
235: 'ea'u-9cea|:u+12'a<c'u-12f:|
236: |:u+11'g<c'u-11d:||:u+12'gb'u-12d:|
237: /[M]
238: u90|:u+9'fa'u-9c:||:u+12'gb'u-12d:|
239: |:u+9'g<c'u-9e:||:u+9'g+b'u-9e:|
240: |:u+13'fa'u-13c:||:u+8'gb'u-8d:|
241: u+14|:'fg<c4':|'eg<c4'&'eg<c16'u-1'g<ce8.'
242: |:'g<cf4':|'g<ce8.''g<ce8.''g<ce8'
243: / [N]
244: o4l8u95|:u+10'ce'u-10>g<:||:u+10'd>b'u-10>g<:|
245: |:4u+10'ce'u-10>a<:||:u+12'cf'u-12>a<:||:u+9'eg'u-9c:|
246: |:u+9'fa'u-9c:||:u+8'cd'u-8>g<:|
247:
248: |:u+10'ce'u-10>g<:||:u+10'd>b'u-10>g<:||:4u+10'ce'u-10>a<:|
249: |:u+12'cf'u-12>a<:||:u+9'eg'u-9c:|
250: |:u+9'fa'u-9d:||:u+12'df'u-12>b<:|
251:
252: |:u+10'ce'u-10>g<:||:u+10'd>b'u-10>g<:|
253: u+10'ce'u-10>a<ec16'ea<c16'u+11'ea<c4''cea4'u-11
254: |:u+9'fa'u-9c:||:u+9'eg'u-9c:||:u+9'fa'u-9d:||:u+9'dg'u-9>b<:|
255:
256: |:2|:u+10'ce'u-10>g<:||:u+10'd>b'u-10>g<:|
257: |:4u+10'ce'u-10>a<:||:u+12'af'u-12>c<:||:u+9'eg'u-9c:|
258: |:u+9'fa'u-9d:||:u+12'dg'u-12>b<:|:| ¥21
259:
260: |:u+10'ce'u-10>g<:||:u+10'd>b'u-10>g<:|
261: |:4u+10'ce'u-10>a<:||:u+12'af'u-12>c<:||:u+9'eg'u-9c:|
262: |:u+9'fa'u-9d:||:u+12'dg'u-12>b<:|
263: /----------- piano(l.h.) ----------------------------------------
264: (t4)
265: l1rrrr |:cc2..>g8<:|l8f4.cf2>g2g4bg|:4<c4.>g:|
266: /[A]
267: <c2..c@d0@d1 c2..>b@d0@d1
268: a2..a@d0@d1 f4.f@d0@d1 g4.g@d0@d1
269:
270: <c2..c@d0@d1 c2..>b@d0@d1
271: a2@d0@d1 f4.f@d0@d1 g4.g@d0@d1 g4.g@d0@d1
272:
273: f4.f@d0@d1 g4.g@d0@d1 <c4.>g@d0@d1 <e4.>b@d0@d1
274: <f4.c@d0@d1 g4.d@d0@d1 |:4c4.>g<@d0@d1 :|
275: / [C]
276: o3l8u100c2..c@d0@d1 c2..>b@d0@d1
277: a2..a@d0@d1 f4.f@d0@d1 g4.g@d0@d1
278:
279: <c2..c@d0@d1 c2..>b@d0@d1
280: a2@d0@d1 f4.f@d0@d1 g4.g@d0@d1 g4.g@d0@d1
281:
282: f4.f@d0@d1 g4.g@d0@d1 <c4.>g@d0@d1 <e4.>b@d0@d1
283: <f4.c@d0@d1 g4.d@d0@d1 c4.>g<@d0@d1 <gc16g.c@d0@d1
284: / [E]
285: o3l8c4.c@d0@d1 >b4.b@d0@d1 a4.a@d0@d1 g4.g@d0@d1
286: <f4.f@d0@d1 e4.e@d0@d1 d4.d@d0@d1 >g4g4@d0@d1
287:
288: o3l8c4.c@d0@d1 >b4.b@d0@d1 a4.a@d0@d1 g4.g@d0@d1
289: f4.f@d0@d1 e4.e@d0@d1 d4.d@d0@d1 g4g4@d0@d1
290: / [F]
291: <c1@d0@d1 c2.>gg+@d0@d1 a1@d0@d1 f2g2@d0@d1
292:
293: <c1@d0@d1 c2..>b@d0@d1 a1@d0@d1 g2<g4g4@d0@d1
294: /[G]
295: >f2@d0@d1 g2@d0@d1 <c2@d0@d1 >e2@d0@d1
296: f2@d0@d1 g2@d0@d1 <c1@d0@d1
297: /[H]
298: o3l8c4.c@d0@d1 >b4.b@d0@d1 a4.a@d0@d1 g4.g@d0@d1
299: <f4.f@d0@d1 e4.e@d0@d1 d4.d@d0@d1 >g4g4@d0@d1
300:
301: <c4.c@d0@d1 >b4.b@d0@d1 a2@d0@d1 g4.f8@d0@d1
302: <f4.f8@d0@d1 e4.e8@d0@d1 d4.d8@d0@d1 >g4.g8@d0@d1
303: /[I]
304: o2l8|:|:a-<e->a-<e->:|@d0@d1 |:a-<d>a-<d>:|@d0@d1:|
305: /[J]
306: <e-4.e-@d0@d1 d4.d@d0@d1 c4.c@d0@d1 >b-4.b-@d0@d1
307: a-4.a-@d0@d1 g4.g@d0@d1 f4.f@d0@d1 b-4.b-@d0@d1
308:
309: <e-4.e-@d0@d1 d4.d@d0@d1 c4.c@d0@d1 >b-4.b-@d0@d1
310: a-4.a-@d0@d1 g4.g@d0@d1 f4.f@d0@d1 b-4b-4@d0@d1
311: /[K]
312: |:3a-4.a-@d0@d1 b-4.b-@d0@d1 <e-4.e-@d0@d1 >g4.g@d0@d1:|
313: a-4.a-@d0@d1 b-4.b-@d0@d1 g1&g1@d0@d1
314: /[L]
315: o3c2..c@d0@d1 c2..>b@d0@d1
316: a2..a@d0@d1 f4.f@d0@d1 g4.g@d0@d1
317: <c2..c@d0@d1 c2..>b@d0@d1
318:
319: a4.a@d0@d1 f4.f@d0@d1 |:g4.g@d0@d1 :|
320: /[M]
321: f4.f@d0@d1 g4.g@d0@d1 <c4.c@d0@d1 e4.e@d0@d1
322: >f4.f@d0@d1 g4.g@d0@d1 <c4g<c@d0@d1 >>cf16g<c.@d0@d1
323: c4.>g@d0@d1 <cc4.@d0@d1
324: / [N]
325: o3l8c4.c@d0@d1 >b4.b@d0@d1 a4.a@d0@d1 g4.g@d0@d1
326: <f4.f@d0@d1 e4.e@d0@d1 d4.d@d0@d1 >g4g4@d0@d1
```

▶受験生はパソコンに投資することができなくて悲しい身分です。3年間バイトを一度もできません。

林　哲司(18)栃木県

```
327:
328: o3l8c4.c@d0@d1 >b4.b@d0@d1 a4.a@d0@d1 g4.g@d0@d1
329: f4.f@d0@d1 e4.e@d0@d1 d4.d@d0@d1 g4g4@d0@d1
330:
331: <c4.c@d0@d1 b4.b@d0@d1 a4.a@d0@d1 g4g4@d0@d1
332: f4.f@d0@d1 e4.e@d0@d1 d4.d@d0@d1 g4g4@d0@d1
333:
334: |:o3l8c4.c@d0@d1 >b4.b@d0@d1 a4.a@d0@d1 g4.g@d0@d1
335: f4.f@d0@d1 e4.e@d0@d1 d4.d@d0@d1 g4g4@d0@d1:| ¥21
336:
337: o3l8c4.c@d0@d1 >b4.b@d0@d1 a4.a@d0@d1 g4.g@d0@d1
338: f4.f@d0@d1 e4.e@d0@d1 d4.d@d0@d1 g4g4@d0@d1
339: /-----------------------------------------------------------------
340: (t5)
341: l1rrrr l16c4.>g&(ga)g4.c&(cd)c4..(cg)&c4c>g<c&(cd)
342: c&(cd)c8&c2.&c2.&(cg8)&c8 f2&f8>a8<c8(ga8)g2>g4&(g<c4)&>g1&g1
343: |:12r1:|r4.o4c8&(c>g)g8.&(g>g4)
344: @e35,30u110o2c1c2..g+8a1f4.ff+g4>g8ag
345: <c2..c8c2..>b8a2f4.f8g2g8.<d&d4
346: >f4.f+8g8g4b8<c4.d8e4.e8f4.ff+g2c4.g&(g<c)&>g2
347: /[E]
348: o2l16c2>b8b4b8a4.a8g8g4<e8f2e8e4ecd4.a8g4>g8b8
349: <c4.gc>b8b4gg+a4.<a8g8g4gef4.cfe8e4ecd4.ff+g8gd>g4
350: /[F]
351: u100o2c4g8&(g<c8)&>g4.<(dc)&du-10(fg8)&f4.u+10g8f8>b4
352: a2..gef8<f8cfr(fg)rdr>b&bgd8
353: c4g8&(g<c8)&>g4.<(ga)(ag) g4..&(g>c16)c4.>(b<a8)&
354: >b4.<<def8.e&e8dc >g8<(cd8)gd(ga)&gbabg&gfed
355: /[G]
356: u110>f4.ff+g8g4>bb<c4.dd+e4>e4 f4.ff+g4.g8 <c4.g&(g<c)&>g2
357: /[H]
358: c4.c8>b4.g8a4.<a8g8g4def4.cfe8e4ecd4.a8g4>g8ab
359: <c4.g<c>>b8b4gg+a4.<a8g8g4def4.cfe8e4cc+d4.ff+g4>g4
360: /[I]
361: o1l8u110|:8a-:|<|:8a-:|>|:8a-:|<a-a-a-a->a-4<a-4
362: /[J]
363: l16e-4.b-e-d8d4>b-b <c4.g<c>b-8b-4b-g
364: a-4.e-a-g8g4ge- f4.<c8>b-4>b-8<f>b-
365: <e-4.b-e-d8d4>b-b <c4.g<c>b-8b-4b-g
366: a-4.<e->a-g8g4e-e f4.a-ab-4>b-b-8b-
367: /[K]
368: a-4.a-ab-8b-4<cd e-4.>ff+g8g8<g8>g8
369: a-4.a-a+b-8b-4<cd e-4.ff+g&(g<b)b<cd8>g&(g>g)
370:
371: a-4.a-ab-8b-4cd e-4.ff+g4>g4 a-4&a-<e-a-a+b-fb-8b-8ff+
372: u-10g2.&g<u-20d8&(dg)&d1
373: /[L]
374: u110|:6r1:|a2f2g4&g8.dgd>g<d>g8g8
375: /[M]
376: >f4.ff+g8g4ab <c4.dd+e4>e4 <f4.ff+g8g(d>g)g4
377: <c4&cg8&(g<c)&>g4&g<g8&(g<c)& >g4&gg8&(gc)&g2
378: /[N]
379: o2c4.g<c>>b8b4gg+ a4.<a8g8g4de f4.<cf>e8e4cc+ d4.ff+g4>g8bg
380: <c4.gc>b8b4gg+ a4.<eagag4de
381: f4.<c>fe8e4cc+ d8<cc+d8>ff+g8>g8g8ab
382: <c4.gc>b8b4gg+ a4.<eag8g4de f4.cfe8e4ec d8cc+d8.<cedr>b8g>ab
383: <c4.gc>b8b4gg+ a4.<eag8g4de
384: f4.<c>fe8e4cc+ d8<cc+d>adag8d8>gabg
385: <c4.g<c>>b8b4gg+ a4.<a8g8g4de
386: f4.<c>fe8e4. d8a<d&d>da8gd8.>g8ab ¥21
387: <c4.gc>b8b4gg+ a8aa&a<a8.gag4de
388: f8f4<cfl12ec>gecc+ l16d4.ff+g8&(g>g8)&<g4
389: /-----------------------------------------------------------------
390: (t6)
391: l1|:6rrrr:|r l8d-e-e-e-e-e-e-e- |:16e-:|
392: e-e-e-e-e-e->g-16g-16b- <|:28e-:|e-e-e-4
393: /[D]
394: d-|:31e-:|
395: /[E]
396: l8d-e-e-e-|:28e-:|d-e-e-e-|:24e-:|e-e-e-4
397: /[F]
398: d-|:27e-:|e-e-r16>g-16b-< d-|:31e-:|
399: /[G]
400: d-|:27e-:|e-e-e-4
401: /[H]
402: |:d-|:31e-:|:|
403: /[I]
404: l8d-|:27e-:|e-2
405: /[J]
406: |:3u+10d-u-10|:31e-:|:| d-e-e-e-|:12e-:|d-e-e-e-u+10d-2
407: d-2r4.u-10l24e-e-e- e-4r2.
408: /[L]
409: l8|:7r1:|>u115b-4<d-u90e-r2
410: /[M]
411: d-e-e-e-|:32e-:|e-2
412: /[N]
413: d-e-e-e-|:28e-:|d-e-e-e-|:24e-:|e-e-r4
414: d-e-e-e-|:28e-:|d-e-e-e-|:20e-:|d-4d-4r2
415: d-e-e-e-|:12e-:|d-e-e-e-e-e-e-e- d-8.d-16r8d-8 r16d-8.d-4 ¥21
416: d-e-e-e-|:12e-:|d-e-d-e-e-6e-12d-e-
417: e-e-e-e-e->g-16g-32g-32g-16g-16r8
418: /-----------------------------------------------------------------
419: (t7)
420: l1|:6rrrr:| l16u+10r4.c64c16..rc>a8a64a32.affu-10
421: o2|:7r2d2:|r4d4d16<c8>a8aff
422: /[D]
423: |:7r4d4:|r4ddrd
424: /[E]
425: o2l16|:14r4d4:|r4d8.d8d8<c8>aaa
426: /[F]
427: |:8r4d4:||:8r4d4:|
428: /[G]
429: |:6r4d4:|r4ddr8ddr<c8>aff
430: /[H]
431: |:15r4d4:|r4dd8d
432: /[I]
433: o2l16r4d8.drd8.dd8d r4d4r8ddrd8. r4d8.d8d8.d8.d
434: rd8.d8.dl24ru+10ddddd<cc>aaffu-10
435: /[J]
436: l16|:28r4d4:|
437: r4d8.drd<c8cc>aar1r1
438: /[L]
439: |:7r1:|r4d4d<c8>a8fff
440: /[M]
441: |:7r4d4:|r4d8.dr4d8.drd8<c32c32c>aff
442: /[N]
443: |:15r4d4:|dd8d<c>a<cc> |:6r4d4:|r4d8.d8d8.dd8d
444: |:4r4d4:|r4d8.d8d8.dd8dr8l24dddrddd<c>adafddddd<cc>aa
445:
446: l16r4d4r4d4 r4d8.drd8.dd8d r4d8.dr4d8.d
447: rl32ddddr16<cccc>r16aa aar16ffffl24rfffff ¥21
448: l16r4d4r4d8.dr4dd8drd8.dd8d r4d8.dr4d8.dr4dd8drd8.r8<cc
449: /-----------------------------------------------------------------
450: (t8)
451: l1|:6rrrr:|r l8o2|:7ccr4r2:|cl16ccr8.cr2
452: /[D]
453: l8|:3c4r4ccr4:|c4r4c16c8c32c32r4
454: /[E]
455: o2l8|:3c4r4ccr4:|c4r4c16c8c16r4|:3c4r4ccr4:|c4r4c4r4
456: /[F]
457: |:3c4r4ccr4:|c4r4c16c8c16r4
458: |:3c4r4ccr4:|c4r4c16c8c16r16c16r16c16
459: /[G]
460: |:3c4r4ccr4:|ccr.c32c32r2
461: /[H]
462: |:3c4r4ccr4:|c4r4c16c8c16r16c16r16c16
463: l8|:3c4r4ccr4:|c4r4c16c8c32c32r4
464: /[I]
465: o2l8ccl16r4c8.c32c32r4c8c8rcrcrc32c32r8c8.c
466: c8ccr4c8ccr4c8ccr4c4r4
467: /[J]
468: |:3c4r4c8c8r4:|c4r4ccrcr8cc |:3c4r4c8c8r4:|c4r4c8ccrcrc
469: /[K]
470: |:3c4r4c8c8r4:|c4r4ccrcr4 |:c4r4c8c8r4:|c4r4c4r4 c4r2.r1
471: /[L]
472: |:7r1:|r8ccr8.cr2
473: /[M]
474: |:3c2c8c8r4:|c2cc8cr4c8.cr4c4r4
475: /[N]
476: l8|:3c4r4ccr4:|c4r4c16c8c16r16c16r16c16
477: l16c4r4c8c8r4c4r4c8ccr8c8 c4r4c8c8r4 c8.cr2.
478: c4r4c8c8r4c4r4c8ccr4c4r4c8c8r4c8ccr4c8.cr4
479: c4r4c8c8r4c4r4c8c8r4c8ccr4c8.c32c32r4c12c24r8c4r2
480: c4r4c8c8r4c4r4c8.c32c32r4c8.cr4c8c8r4c8.cr8c8rc8.c4 ¥21
481: c4r4c8c8r4c8ccr4c8.c32c32r4c8ccr4c8c8r4ccrcr4c2
482: /-----------------------------------------------------------------
483: (t9)
484: l1|:6rrrr:|r
485: l16o2c1c2..g+8a1f4.ff+g4>g8ag
486: <c2..c8c2..>b8a2f4.f8g2g8.<d&d4
487: >f4.f+8g8g4b8<c4.d8e4.e8f4.ff+g2c4.g&(g<c)&>g2
488: /[E]
489: o2c2>b8b4b8a4.a8g8g4<e8f2e8e4ecd4.a8g4>g8b8
490: <c4.gc>b8b4gg+a4.<a8g8g4gef4.cfe8e4ecd4.ff+g8gdg4
491: /[F]
492: u70o2c4g8&(g<c8)&>g4.<(dc)&du-10(fg8)&f4.u+10g8f8>b4
493: a2..gef8<f8cfr(fg)rdr>b&bgd8
494: c4g8&(g<c8)&>g4.<(ga)(ag) g4..&(g>c16)c4.>(b<a8)&
495: >b4.<<def8.e&e8dc >g8<(cd8)gd(ga)&gbabg&gfed
496: /[G]
497: u80>f4.ff+g8g4>bb<c4.dd+e4>e4f4.ff+g4.g8<c4.g&(g<c)&>g2
498: /[H]
499: c4.c8>b4.g8a4.<a8g8g4def4.cfe8e4ecd4.a8g4>g8ab
500: <c4.g<c>>b8b4gg+a4.<a8g8g4def4.cfe8e4cc+d4.ff+g4>g4
501: /[I]
502: o1l8u90|:a-a-a-a-:|<|:a-a-a-a-:|>|:a-a-a-a-:|<a-a-a-a->a-4<a-4
503: /[J]
504: l16e-4.b-e-d8d4>b-b <c4.g<c>b-8b-4b-g
505: a-4.e-a-g8g4ge- f4.<c8>b-4>b-8<f>b-
506: <e-4.b-e-d8d4>b-b <c4.g<c>b-8b-4b-g
507: a-4.<e->a-g8g4e-e f4.a-ab-4>b-b-8b-
508: /[K]
509: a-4.a-ab-8b-4<cd e-4.>ff+g8g8<g8>g8
510: a-4.a-a+b-8b-4<cd e-4.ff+g&(g<b)b<cd8>g&(g>g)
511:
512: a-4.a-ab-8b-4cd e-4.ff+g4>g4 a-4&a-<e-a-a+b-fb-8b-8ff+
513: u-10g2.&g<u-20d8&(dg)&d1
514: /[L]
515: u80|:6r1:|r1d4&d8.dgd>g<d>g8g8
516: /[M]
517: >f4.ff+g8g4ab <c4.dd+e4>e4 <f4.ff+g8g(d>g)&<d4
518: c4&cg8&(g<c)&>g4&g<g8&(g<c)& >g4&gg8&(gc)&g2
519: /[N]
520: o2c4.g<c>>b8b4gg+ a4.<a8g8g4de f4.<cf>e8e4cc+ d4.ff+g4>g8bg
521: <c4.gc>b8b4gg+ a4.<eagag4de
522: f4.<c>fe8e4cc+ d8<cc+d8>ff+g8>g8g8ab
523: <c4.gc>b8b4gg+ a4.<eag8g4de f4.cfe8e4ec d8cc+d8.<cedr>b8g>ab
524: <c4.gc>b8b4gg+ a4.<eag8g4de
525: f4.<c>fe8e4cc+ d8<cc+d>adag8d8>gabg
526: <c4.g<c>>b8b4gg+ a4.<a8g8g4de
527: f4.<c>fe8e4. d8a<d&d>da8gd8.>g8ab ¥21
528: <c4.gc>b8b4gg+ a8aa&a<a8.gag4de
529: f8f4<cfl12ec>gecc+ l16d4.ff+g8&(g>g8)g4
530: /-----------------------------------------------------------------
531: (t10)
532: l1|:9rrrr:|r l1|:7rrrr:| l1rrr
533: o4 r2.c8d8 l16e-4(a-b-)&a-8.b-a-g8a-8l24ga-g
534: f8e-fe-d8c8l16@q1>b-<cd(e-f)e-dc>b-
535: @q0b-a-8.<c8>b-8a-gfe-fga-b-
536: l8<c>b-16r16<e-d16r16gf16r16<c>b-16r16
537:
538: l16<e-4.fe-d8.>g<dr32e-f. e-dc4dc>b-4<(fg32)&f16.fe-
539: >a-4.(fa-)&fg4f@q1ga-b-@q0<c>a-b-<c
540: d>b-<cde-cde-(a-b-)ga-b-
541: /[K]
542: a-2b-2>g4e-8>b-8b8<d8f8a-8
543: gfe-4.d8e-8f8a-8 g4<b-8l24a-b-a-ga-gfgfe-fe-de-d
544: l16c4(fg)&f8.f8g8a-rb-a-g4f8g8>b<dfa-gfe-f
545: e-fga-b-ga-b-<c>a-b-<cd>b-<cd
546: c2.u-24(cd32)&c8&u-10(cc)&c32>b1
547: /[L]
548: @p64@e60,30 @v90|:13r1:|
549: /[N]
550: u70|:5o3l2c>bag<fed>g:| ¥21 o3l2c>bag<fed>g
551: /-----------------------------------------------------------------
552: (t11)
553: l1|:9rrrr:|r l1|:7rrrr:| l1rrr
554: o4 r2.c8d8 l16e-4(a-b-)&a-8.b-a-g8a-8l24ga-g
555: f8e-fe-d8c8l16@q1>b-<cd(e-f)e-dc>b-
556: @q0b-a-8.<c8>b-8a-gfe-fga-b-
557: l8<c>b-16r16<e-d16r16gf16r16<c>b-16r16
558:
```


▶とうとうMIDI楽器(SC-88)を買いました。やはり，MIDIは音が違いますね。しかし，「餓狼SP」を買ったらハードディスクの必要性を痛切に感じ始めた今日この頃です。
高木　奈津彦(22)長野県

```
559: l16<e-4.fe-d8.>g<dr32e-f. e-dc4dc>b-4<(fg32)&f16.fe-
560: >a-4.(fa-)&fg4f@q1ga-b-@q0<c>a-b-<c
561: d>b-<cde-cde-(a-b-)ga-b-
562: /[K]
563: r4>g4f8g8a-8b-8g4e-8>b-8b8<d8f8a-8
564: gfe-4.d8e-8f8a-8 g4<b-8l24a-b-a-ga-gfgfe-fe-de-d
565: l16c4(fg)&f8.f8g8a-rb-a-g4f8g8>b<dfa-gfe-f
566: e-fga-b-ga-b-<c>a-b-<cd>b-<cd
567: c2.u-24(cd32)&c8&u-10(cc)&c32>b1
568: /[L]
569: @p64@e60,30 @v90|:13r1:|
570: /[N]
571: u70|:5o3l2c>bag<fed>g:| ¥21 o3l2c>bag<fed>g
572: /---------------------------------------------------------------
573: (t12)
574: l16|:11r1:|r2..g8
575: <e8e8e8ee&e4>g8g8<d8d8d8dd&d4r4r8c8c8cc8.>b8a8b<c&c4d4c4>b4
576: <e8e8e8ee&e4>rggg<d8d8d8dd&d4r2>a8a8<a8a8a8ag&g4r8d8e4d4
577:
578: c4r8e8d8.c8.>b8<c4r8d8e4d4c4r8e8d4c8>b<c&c2r2r2..>gg
579: u110<e8e8e8ee&e4>r8gg<d8d8d8dd&d4r4r8c8c8cc8.>b8a8b<c&c4d4c4>b4
580: <e8e8e8ee&e4>rggg<d8d8d8dd&d4r2>r8aa<a8a8a8ag&g4r8d8e4d4
581: c4r8dde8d8c8>b<c&c4r8d8e4d4c4r8e8d4c8>b<c&c4r2.
582:
583: e2.b4<c4r2dc8>b<c4r2dc8>b<c4r8d8c4>b4
584: e4e4r4b4<c4r2dc8>b<c4r2dc8>b<c2>b2
585:
586: |:7r1:|r2e4d4
587: c4r8e8d4c8>b<c&c4r8d8e4d4c4r8e8d4c8>b<c&c4r2.
588: e2.b4<c4r2dc8>b<c4r2dc8>b<c4r8d8c4>b4
589: e4e4r4b4<c4r2dc8>b<c4r2dc8>b<c2>b2
590: u125<c2r8>a-8b-8<c8>b-1<c2r8c8d8e-8d4r4d4d4
591: y$63,$01 y$62,$66 y$06,$60 e-1u90|:15r1:|y$06,$40 r2...>>g
592:
593: <e4e8.f8.e8r8>gg<d8d8d8dd&d4r4
594: r8c8c8cc8.>b8a8b8<c4d4c4>b4
595: <e8e8e8ee&e4>rggg<d8d8d8dd&d4r2> u110r8aa<a4a8a8g4r8d8e4d4
596: c4r8de8.d8c8>b<c&c4r8d8e4d4c4r8e8d4c8>b<c&c4r2.r1
597:
598: e2.b4<c4r2dc8>b<c4r2dc8>b<c4r8d8c4>b4
599: e4e4r4b4<c4r2dc8>b<c4r2dc8>b<c2>b4r4
600: e2.b4<c4r2dc8>b<c4r2dc8>b<c4c4>b4r4
601: e2.b4<c4r2dc8>b<c4r2dc8>b<c4.c8>b4r4
602: e2.b4<c4r2dc8>b<c4r2dc8>b<c4r8d8c4>b4 ¥21
603: e4e4r4b4<c4r4..d8c8>b<c4r4..d8c8>b<c2>b4r4
604: /---------------------------------------------------------------
605: (t13)
606: l8|:25r1:|cccc16c16&c4r4r1reee16e.dcd16e16&e4f4e4d4
607: r1r1r1r1 r2.ef16g16&g4rgg+4g+4a4rag4gg16g16&g4r2. |:49r1:|
608: c4c.d.cr4|:7r1:|r2.ef16g16&g4rgg+4g+4a4rag4gg16g16&g4r2.
609: |:9r1:| c2.g4a4r2b16ag16a4r2b16ag16a4a4g4r4
610: c2.g4a4r2b16ag16a4r2b16ag16a4.ag4r4
611: c2.g4a4r2b16ag16a4r2b16ag16a4rba4g4 ¥21
612: c4c4r4g4a4r4..bag16a4r4..bag16a2g4r4
613: /---------------------------------------------------------------
614: (p)
```

リスト5　ENDLESS RAINのカウンタ表示

```
 1:00005C40 00000000   2:00005C40 00000000   3:00005C40 00000000   4:00005C40 00000000
 5:00005C40 00000000   6:00005C40 00000000   7:00005C40 00000000   8:00005C40 00000000
 9:00005C40 00000000  10:00005C40 00000000  11:00005C40 00000000  12:00005C40 00000000
13:00005C40 00000000
```

リスト6　ファイナルファンタジーV

```
  1:
  2: / ---------------------------------------------------
  3:
  4: .comment ＜Final Fantasy 5＞ レナのテーマ  by め－おさ
  5:
  6: /                         作曲：植松 伸夫(SQUARE)
  7:
  8: / ---------------------------------------------------
  9:
 10: (i)(o82)
 11:
 12: (m01,2048)(aMIDI01,01)
 13: (m02,2048)(aMIDI02,02)
 14: (m03,2048)(aMIDI03,03)
 15: (m04,2048)(aMIDI04,04)
 16: (m05,2048)(aMIDI05,05)
 17: (m06,2048)(aMIDI06,06)
 18: (m07,2048)(aMIDI07,07)
 19: (m08,2048)(aMIDI08,08)
 20: (m09,2048)(aMIDI09,09)
 21:
 22: / --- SC-55 SET ----------------------------------
 23:
 24: .Roland_exclusive $10,$42 ={$40,00,$7F,00}
 25:
 26: .SC55_reverb $10 ={3,3,0,68,68,0,9}
 27:
 28: .SC55_chorus $10 ={0,0,20,9,10,5,14,18}
 29:
 30: / ---------------------------------------------------
 31:
 32:
 33: / --- FLUTE ---
 34:
 35: (t1)@i$41,$10,$42r16@v125@74y91,100y93,10r16r1r1@u100o5c8&@m30c8@m
 36: (t1)g8&@m30g8@mf8&@m30f8@me8&@m30e8@md8e8>b8&@m30@q14b2^8@mq8<d8e8
 37: (t1)>b8&@m30b4@ma8b8<d8c8>a8&@m30@q10a2.@mq8d8e8a8&@m30a4@me8d8e8
 38: (t1)b8&@m30@q12b2..@mq8d8e8b8&@m30b4@mg8e8g8<c+8&@m30@q4c+2^8@mq8>a8b8
 39: (t1)[do]
 40: (t1)@u100o5c8&@m30c8@mg8&@m30g8@mf8&@m30f8@me8&@m30e8@md8e8>b8&@m30
 41: (t1)@q14b2^8@mq8<d8e8>b8&@m30b4@ma8b8<d8c8>a8&@m30@q10a2.@mq8d8e8
 42: (t1)a8&@m30a4@me8d8e8b8&@m30@q12b2..@mq8d8e8b8&@m30b4@mg8e8g8a8&@m30
 43: (t1)a2..@m|:4r1:|
 44: (t1)[loop]
 45:
 46: / --- CELESTA ---
 47:
 48: (t2)@i$41,$10,$42r16@v115@p44@9y91,127y93,10o7
 49: (t3)@i$41,$10,$42r16@v105@p44y99,1y98,100y6,70y99,1y98,102y6,70@10y91,127y9
3,10o6
 50: (t2,3)r16|:10r1:|
 51: (t2,3)[do]
 52: (t2,3)|:8r1:|@u100@q28g4@q4>b8<c8@q52d4.@q4a8@q76g2d+2@q28g4
 53: (t2,3)@q4>b8<c8@q28d4a4@q172g1
 54: (t2,3)[loop]
 55:
 56: / --- MUSICBOX ---
 57:
 58: (t4)@i$41,$10,$42r16@v115@p40@11y91,100y93,10r16@u100@q4o4f8<e8a8
 59: (t4)@q100<c2^8@q4>>f8<d8g8@q100b2^8@q4>f8<c8e8@q100a2^8@q4>f8b8<d8
 60: (t4)@q100g2^8@q4>e8b8<e8@q100g2^8@q4>a8<c8e8@q100a2^8@q4>f8<c8e8@q100a2^8
 61: (t4)@q4>f8b8<d8@q100g2^8@q4>e8b8<e8@q100g2^8@q4>a8<c+8e8@q100a2^8
 62: (t4)[do]
 63: (t4)@u100@q4o4f8<c8e8@q100a2^8@q4>f8b8<d8@q100g2^8@q4>e8b8<e8@q100g2^8
 64: (t4)@q4>a8<c8e8@q100a2^8@q4>f8<c8e8@q100a2^8@q4>f8b8<d8@q100g2^8@q4>e8
 65: (t4)b8<e8@q100g2^8@q4>f8a8<c8@q100f2^8@q4>c8g8<c8e8f+8d8>a8f+8c8a+8
 66: (t4)<d+8g8a8f8c8>a8c8g8<c8e8f8d8>a+8f8c8d+8g+8<c8@q76d+2
 67: (t4)[loop]
 68:
 69: / --- STRINGS 1 ---
 70:
 71: (t5)@i$41,$10,$42r16@v127@p74@50y99,1y98,99y6,60y91,100y93,10r16
 72: (t5)|:10r1:|
 73: (t5)[do]
 74: (t5)@u100o5e1&e2d2&d1c2d2e1&e2d2&d1c1c1c1c2d2g1
 75: (t5)[loop]
 76:
 77: / --- STRINGS 2 ---
 78:
 79: (t6)@i$41,$10,$42r16@v127@p85@50y99,1y98,99y6,60y91,100y93,10r16
 80: (t6)|:9r1:|r2@u100o4a2
 81: (t6)[do]
 82: (t6)@u100o4a1g1&g1a1a1g1&g1a1e2f+2g2f2e2f2<c1
 83: (t6)[loop]
 84:
 85: / --- STRINGS 3 ---
 86:
 87: (t7)@i$41,$10,$42r16@v127@p104@50y99,1y98,99y6,60y91,100y93,10r16
 88: (t7)|:9r1:|r2@u100o4e2
 89: (t7)[do]
 90: (t7)@u100o4c1>b1&b1a2b2<c1>b1&b1a1g2a2a+2a2g2a2<d+1
 91: (t7)[loop]
 92:
 93: / --- BASS ---
 94:
 95: (t8)@i$41,$10,$42r16@v118y99,1y98,100y6,70y99,1y98,102y6,78@36y91,32y93,10
 96: (t9)@i$41,$10,$42r16@v110y99,1y98,100y6,114y99,1y98,102y6,78@33y91,32y93,10
 97: (t8,9)r16|:10r1:|
 98: (t8,9)[do]
 99: (t8,9)@u120o1f1f1e1a8<e8@b0a16.&@b-128a32&@b-256a64&@b-384a64&@b-512
100: (t8,9)a96&@b-640a*1&@b-768a96&@b-896a*1&@b-1024a*1&@b-1152a*1&@b-1408
101: (t8,9)a*1&@b-1536a*1&@b-1792a*1&@b-2048a*1&@b-2176a*1&@b-2560a*1&@b-2816
102: (t8,9)a*1&@b-3200a*1&@b-3584a*1&@b-4224a*1@b0>g8<d8@b0g16.&@b-128g32&
103: (t8,9)@b-256g64&@b-384g64&@b-512g96&@b-640g*1&@b-768g96&@b-896g*1&@b-1024
104: (t8,9)g*1&@b-1152g*1&@b-1408g*1&@b-1536g*1&@b-1792g*1&@b-2048g*1&@b-2176
105: (t8,9)g*1&@b-2560g*1&@b-2816g*1&@b-3200g*1&@b-3584g*1&@b-4224g*1@b0
106: (t8,9)>f1f1e1f1<c2..c8c2..c8c2>a+2g+1
107: (t8,9)[loop]
108:
109: (p)
```

リスト7　ファイナルファンタジーVのカウンタ表示

```
1:00000798 00000900   2:00000798 00000900   3:00000798 00000900   4:00000798 00000900
5:00000798 00000900   6:00000798 00000900   7:00000798 00000900   8:00000798 00000900
9:00000798 00000900
```

▶EXEディスクを申し込むことにした。……あ，宛名が長すぎる！
井村　英二(23)滋賀県

(善)のゲームミュージックでバビンチョ

西川善司

●MIDI POWER Ver.5.0　SNATCHER
CD:KICA-7645　2,800円(税込)
キングレコード　10/21発売

コナミMIDI POWERシリーズも5作目。今回はMSX2, PC-8801, PCエンジン(CD-ROM²)用に発売されて爆発的人気を博したサイバーパンクアドベンチャー「スナッチャー」だ。演奏音源を従来のSC-55からSC-88へパワーアップ。SC-88を買い渋っている人はこのCDを聴いてから決断するのもいいかもしれない。「スナッチャー」といえば音楽もコナミの「カッコイイ」どころの曲が勢ぞろいしているワケで，それがSC-88で聴けちゃうなんて，ファンには期待するなっていうほうが無理。曲のアレンジは最高だし，SC-88ならではのサウンド(拡張されたドラムサウンドキットなど)も楽しめる文句なしの1枚。こりゃ買いだ。ところでコナミさん,「スナッチャー」をX68000に移植してくださいな。

お勧め度　10

●ナムコ　ゲームサウンド　エクスプレス　Vol.16　ATTACK OF THE ZOLGEAR
CD:VICL-15035　1,500円(税込)
ビクターエンタテインメント　10/21発売

ナムコの「ギャラクシアン3」の流れを汲む最新大型3Dシューティングゲーム「アタック・オブ・ゾルギア」。私はプレイしたことがないが，聞くところでは映像のリアリティはもはや実写と見間違うほどのレベルとか。「ギャラクシアン3」の後継ということでBGMの至るところにそのフレーズが織り込まれている。CD後半にはゲームの臨場感を再現するショートシナリオが，英語版と日本語版両方にて収録されている。

お勧め度　8

●ラグナレック
CD:DPCX-5045　2,500円(税込)
データム・ポリスター　10/26発売

グローディアのパソコンゲーム「ラグナレック」の全曲集。全曲，本格的な管弦楽アレンジがされており，その演奏の重厚さには圧倒される。1曲1曲が絵画のようなメッセージを持ち，全体として物語性のある交響組曲が構成されている。ヒロイン「フィリア」のイメージガール「優加しおり」が歌うイメージソングシングルアルバム(DPDX-5013，1,000円(税込))も同時発売。

お勧め度　8

●ワールドヒーローズ2ＪＥＴ
VHS:GV-010　6,500円(税込)
新声社　発売中

新声社が「ワールドヒーローズ2JET」のビデオを発売。先月号で紹介したスーパーストIIXのビデオは，連続技や対戦攻略が中心の攻略ビデオだが，このビデオは他社が出しそうな普通のゲームプローモーションビデオになってしまっている。ただ，さりげなくCPU戦のパターン・ハメ方法が収録されており，辛うじてゲーメストフレーバーが感じられる。収録時間は70分。

お勧め度　7

番外

神奈川県の渡邊孝行さんからのリクエストを紹介します。

●PMD98
CD:NACL1087　2,800円(税込)
NECアベニュー　発売中

PC-9801を4台にFM音源ボード6枚，PCM音源ボード2枚，MIDI I/Fボード4枚を装備させ多重録音したパソコンミュージックのCDアルバム。タイトルの「PMD98」とはPC-9801の世界では最もメジャーなミュージックドライバの名前で，このCDはこのドライバの演奏を収録している。収録曲はオムニバス形式でY.M.O.の「FIRECRACKERS」「RYDEEN」をはじめとしたコピー曲がメイン。データ制作者はガイナックスの「プリンセスメーカー」「サイレントメビウス」などの音楽を担当した梶原正裕氏，「ニューラルギア」「スタートレーダーX68000」などの音楽を担当した与猶啓至氏ら。彼らデータ制作者のオリジナル曲も収録されている。

ゴージャスな超多チャンネル4OPのFM音源サウンドは，何か不思議な印象。ただ，曲によってはアンビエンス系のエフェクトが皆無で，「不思議な印象」よりも「チープ」さを感じてしまうことも。

お勧め度　8

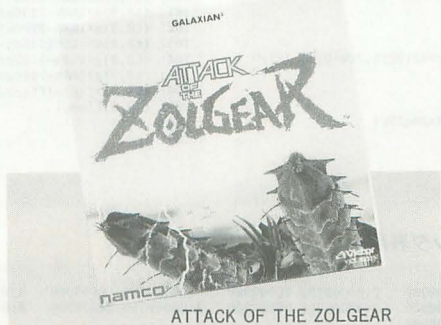
●Kaleidoscope
CD:UCD1032-1033
3,000円(送料・手数料込)
MYU-RECORDINGS　発売中

MYU-NETという草の根のパソコン通信ネットワークで活躍中のアマチュア&プロのミュージシャンたちによるCDが善バビ宛てに送られてきた。上記の「PMD98」にも参加している与猶啓至氏やOh!Xのイラストレーター「てるてる」こと高橋哲史氏の書き下ろしの曲を収録している。CDは2枚組。アルバムのテーマは「色」，そしてアルバムタイトルは「Kaleidoscope/万華鏡」だ。ボーカル曲にナニなのもあるが，これがアルバムに「ごった煮」的なコク深い味を出しているという説もなきにしもあらず。「色」というテーマのせいか全体的に幻想的で環境音楽的な傾向のモノが多く，聴き手によって好みが分かれるかも。収録曲は全部で34曲，演奏時間にして2時間。

お勧め度　8

なお，このCDは通信販売のみで購入できる。値段は送料手数料込みで3,000円，送金方法は郵便書留もしくは郵便小為替で。
宛先は，〒173　東京都板橋区双葉町35-5
米山正晃　まで

なお，Oh!Xとは一切関係ないので，編集部への問い合わせはご遠慮ください。

SNATCHER

ATTACK OF THE ZOLGEAR

(哲)のショートプロぱーてぃ――その62

代打稼業も楽じゃない

Takahashi Tetsushi 高橋 哲史

10月号にもあったとおり今月は(で)氏はお休みです。そこで代わって登場するのは毎月のイラストでおなじみの(哲)氏です。さて，どうなりますことやら。今月のプログラムはゲームが2本にユーティリティが1本，久しぶりにX1用も登場です。

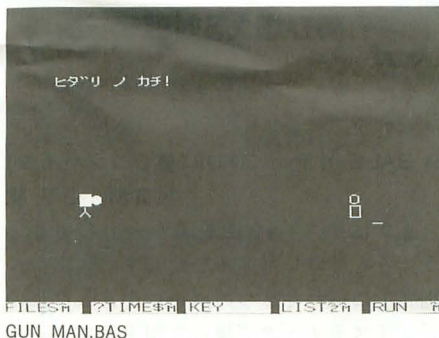
illustration：T.Takahashi

どうもこんにちはー！ 古村聡Jr.FXでぇーすっ……というのはウソです。えー，実はいつもならイラストだけが担当の私なんですが，先月号をお読みになった方はご存じのとおり古村さんが突然入院されることになったので急遽代打を仰せつかったわけです。いやー，プログラムなんて久しぶりなんでいつになく緊張してたりしますが，今月採用された方々は犬にでもかまれたと思ってあきらめてくださいね(おいおい)。というのは冗談で，一度引き受けたからにはこの高橋哲史，命に代えても使命をまっとういたしますんでぜひよろしく！ ただ気合は無限大でも能力値には限界がありますんでそこんとこもよろしく(おい)。

問答無用のX1攻撃!

さていきなりですが1年半ぶりにX1用プログラムの登場です。文句はいわせません。なんといっても私は「LOVE X1」なのです。さらにいえば「X1命」なのです。さらにいえば……（編注：もーいーってば)。失礼しました，少し興奮してしまったようです。気を取り直して冷静に進めてまいりましょう。まずはゲームです。

GUN_MAN.BAS for X1
(要CZ-8FB01，ジョイスティック)
東京都 北浦暁光

まずリスト1を間違いなく入力したあと，RUNで実行してください。ゲームの内容は西部劇よろしくガンマンになって撃ち合いをし，弾が当たったほうの負けという実にわかりやすいゲームになっています。2人用でジョイスティックが2本必要ですが，STICK(2)をSTICK(0)にすればジョイスティック1本とテンキーで遊べます。そーいえば最近映画のほうでも「ワイアットアープ」とか「許されざる者」とかちょっとした西部劇のリバイバルが起きてますよね。んー，時代の流行に敏感なショートプロ(ちょっと遅いか？)。

GUN_MAN.BAS

さてプログラムはオーソドックスにまとまっているんですが，判定部分がやや冗長だったので310行以下を論理式を使ってコンパクトにまとめてみました(リスト2)。こうするとドキュメント性は低くなってしまうので善し悪しなんですが，プログラムは8行ほど縮まります。テクのひとつとして覚えておけばなにかと有用ではないでしょうか？ ちなみに論理式とは，たとえば(S1＝1)という式だった場合，S1＝1(真)ならば－1，S1<>1(偽)ならば0の値をもつというものです。コンピュータの世界ではX1のBASICに限らず真＝－1，偽＝0と扱われることがままあるので覚えておいて損はないでしょう。

しかし単純ながら対戦でやるとこれが結構燃えるんですよね。弾の相殺とかを考慮すればさらに面白くなったかもしれません(いまのままでも十分面白いですけど)。ショートプロの真髄ここにありって感じでしょうか。

ちなみに余談ですがうちのX1turboZはまだ現役で頑張ってもらってます。といっても主に画像取り込み専用なんですが……(珍しく自分でプログラムを組んでX68000に画像を持ってこられるようにしてあるんで)。これはなにについてもいえることだと思いますが，使われてるうちは一生現役なんだと思います。どんな機種にも終わりがきますが，それを決めるのはやはり使っている私たちのほうだと痛感する今日この頃です。

簡易アートギャラリーだ!

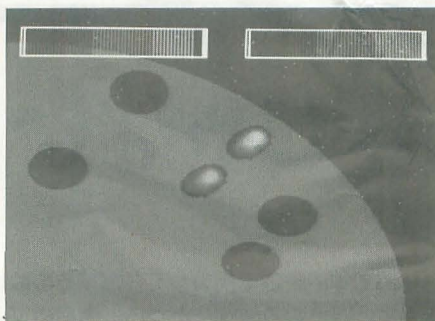
続きましてはグラフィック関連でちょっと小粋なユーティリティをご紹介しましょう。

PIC_MENU.BAS for X-BASIC
(要DIR.FNC，APIC.FNC)
岩手県 佐々木崇

皆様はあふれるPICファイルの整理に悩んだことがおありでしょうか？ 私は大ありです。人からもらったり，通信でダウンロードしたり，あげくの果てに自分で描いたりと収拾のつかなくなったPICファイルの山，山，山。1枚ずつ表示して中身を確認するのも大変だし，かといってこのまま死蔵させるには忍びない。そう，そんなときこそこのPIC_MENU.BASの出番です。

まずはリスト3をエディタなりX-BASIC上なりで打ち込んでください。実行には1992年6月号の付録「10周年記念PRO-68K」に収録されているAPIC.FNCとDIR.FNCが必要なので注意してくださいね。おもむろに実行するとPICファイルのあるディレクトリを聞いてきます。そこで希望のディレクトリ名を打ち込むとあら不思議，そのディレクトリ上にあるPICファイルをくまなく縮小表示したメニューが出来上がるのです。んー，これは便利。もちろんメニューからPICを呼び出して鑑賞するのもカーソルキーとリターンキーでおちゃのこさいさいです。これで整理の苦労が格段に軽減されるというものです。

しかしちょっとしたアイデアでこういう便利なユーティリティができてしまうんで

すね。思わず感心してしまいました。テキストをワークに使ってるのが丸見えだったり，縮小のときの2段階表示にはいかにも手作りの味が感じられてほほえましいですが。それにしてもPIC_MENU.BASは大いに拡張しがいのあるプログラムといえるのではないでしょうか？　現状だと256以上のPICファイルが扱えないのでそれを増やしたりとか，指定ディレクトリだけでなく指定ディレクトリ以下のPICファイルを全部参照するようにしてしまうとか。また生成されるpmiファイルもベタでなく圧縮して持てるようになればかなり容量の節約になると思いますし，メニュー選択画面でのPICファイルのコピー，削除，移動なんかできればこれはもう史上最強のPICユーティリティになること間違いなしです。ちょっと考えただけでもこれだけ拡張案が出てくるのですから，もう「あなたの好みでお料理し・て・ね☆」ってなもんです。人のプログラムの解析，拡張はかなり勉強になりますから，ぜひやってみることをお勧めします。

B_BALL.X

再び燃える対戦ゲーム

さてトリをとるのはまたもや対戦型のゲームです。

B_BALL.X for X68000（要Cコンパイラ）
東京都　小平 覚

まずはリスト4を注意深く入力しましょう。そしてコンパイルしたらすぐに実行。画面上に現れた2つのボールをそれぞれのジョイスティックで操って相手をフィールド外に押し出してしまいましょう。ジョイスティックの向きで移動方向が変わり，Aボタンで高速移動ができます。そしてBボタン＋方向で自分の前に穴を開けることができるのです。巧みな動きで相手を幻惑し，勝利を手にする……。んー，そんな華麗な戦いを堪能したいですね。あ，戦いの最中でも画面上部のエネルギーゲージはちゃんとチェックしておきましょうね。

いやー，なかなかすごいですねー。基本的には「モトス」っぽいゲームなのですが，ボールがうにょうにょ変形したり，地面に穴が開けられたりとなかなか芸の細かさを見せつけてくれます。これが212行で収まってるんですから大したもんです。んー，さすがに常連さんは一味違いますね。とそこで，ふと投稿原稿を見ると「ほんとはテスト前で時間がないはずなんですけど……完成してしまいました」とのお言葉が。んー，よくわかりますねぇ。私も学生時代そうでしたし，いまこうしてライターもどきになっても締め切り間際まで現実逃避をしてたりするし(おいおい)。ま，人間追い詰められないと真の力を発揮できないということで。

いやー，それにしてもプログラム関係の原稿書くのは久しぶり……いやもしかしたら初めてだったかもしれないので(そういえば私がOh!Xで発表したのって音楽プログラムだけだったかも)えらくエネルギーを使ってしまいました。いつもおちゃらけてるようでも古村さんやっぱり偉かったんだなあ……(しみじみ)。とりあえずクモの巣が張った脳味噌をフル回転させて書きましたんで，いたらないところもあったと思いますけどどうかご容赦ください。古村さんが復帰されるまで私もしっかり勉強して代役を務めさせていただきます。みなさんも「古村さんのいないショートプロなんてぇー」などといわずによろしくお願いしますね。もうホント，大変なんすから。それでは皆さん，機会があれば来月お会いしましょう。古村さんも私の脳味噌がオーバーヒートする前に帰ってきてくださいねー。

リスト1　GUN_MAN.BAS

```
10 'ｼｮｷ ｾｯﾃｲ(ｿﾝﾅ ﾀｲｿｳ ﾅ ﾓﾉ ｼﾞｬ ﾅｲﾈ!)
20 INIT:CLS:WIDTH 40:X=1:Y=5:T1=0:T2=0
30 Y$=CHR$(29)+CHR$(29)+CHR$(29)+CHR$(29)+CHR$(31)
40 DIM A$(7):DIM N(7)
50 'ｷｬﾗ ｾｯﾃｲ (ﾆｭｳﾘｮｸ ｽﾙ ﾄｷ ｸｳﾊｸ ﾉ ｶｽﾞ ｶﾞ ﾜｶﾘﾆｸｲﾝﾀﾞ ﾖﾈ)
60 A$(1)="  ● "+Y$+"  ■ "+Y$+"  人 "
70 A$(0)="  ■●"+Y$+"  人 "+Y$+"    "
80 A$(2)="    "+Y$+"  ■●"+Y$+"  人 "
90 A$(3)="    "+Y$+"    "+Y$+" 人■●"
100 A$(4)="○□  "+Y$+" 人  "+Y$+"    "
110 A$(5)=" ○  "+Y$+" □  "+Y$+" 人  "
120 A$(6)="    "+Y$+"○□  "+Y$+" 人  "
130 A$(7)="    "+Y$+"    "+Y$+"○□人 "
140 N(0)=15:N(1)=16:N(2)=16:N(3)=17
150 N(4)=15:N(5)=16:N(6)=16:N(7)=17
160 LOCATE 5,15:PRINT A$(X):LOCATE 29,15:PRINT A$(Y)
170 'ﾒｲﾝ ﾙｰﾁﾝ (IF - THEN - ﾉ ｱﾗｼ ﾀﾞｧｧｧｧｧｰｰｰｰｰ)
180 S1=STICK(1):S2=STICK(2)R1=STRIG(1):R2=STRIG(2)
190 IF S1=8 AND X<>0 THEN X=X-1:LOCATE 5,15:PRINT A$(X)
200 IF S1=2 AND X<3 THEN X=X+1:LOCATE 5,15:PRINT A$(X)
210 IF S2=8 AND Y<>4 THEN Y=Y-1:LOCATE 29,15:PRINT A$(Y)
220 IF S2=2 AND Y<7 THEN Y=Y+1:LOCATE 29,15:PRINT A$(Y)
230 IF R1=-1 AND T1=0 THEN T1=1:Y1=N(X):X1=9
240 IF R2=-1 AND T2=0 THEN T2=1:Y2=N(Y):X2=26
250 FILES=1 THEN LOCATE X1+1,Y1:PRINT "  -":X1=X1+2:IF X1>29 THE
N T1=2
260 IF T1=2 THEN GOSUB 310
270 IF T2=1 THEN LOCATE X2-1,Y2:PRINT "-  ":X2=X2-2:IF X2<7 THEN
 T2=2
280 IF T2=2 THEN GOSUB 400
290 GOTO 180
300 'ｱﾀﾘ ﾊﾝﾃｲ(ﾃﾇｷ)
310 IF Y1=15 AND Y=4 THEN LOCATE 5,5:PRINT"ﾋﾀﾞﾘ ﾉ ｶﾁ!":GOTO 490
320 IF Y1=15 AND Y=5 THEN LOCATE 5,5:PRINT"ﾋﾀﾞﾘ ﾉ ｶﾁ!":GOTO 490
330 IF Y1=16 AND Y=5 THEN LOCATE 5,5:PRINT"ﾋﾀﾞﾘ ﾉ ｶﾁ!":GOTO 490
340 IF Y1=16 AND Y=6 THEN LOCATE 5,5:PRINT"ﾋﾀﾞﾘ ﾉ ｶﾁ!":GOTO 490
350 IF Y1=16 AND Y=4 THEN LOCATE 5,5:PRINT"ﾋﾀﾞﾘ ﾉ ｶﾁ!":GOTO 490
360 IF Y1=17 AND Y=5 THEN LOCATE 5,5:PRINT"ﾋﾀﾞﾘ ﾉ ｶﾁ!":GOTO 490
370 IF Y1=17 AND Y=6 THEN LOCATE 5,5:PRINT"ﾋﾀﾞﾘ ﾉ ｶﾁ!":GOTO 490
380 IF Y1=17 AND Y=7 THEN LOCATE 5,5:PRINT"ﾋﾀﾞﾘ ﾉ ｶﾁ!":GOTO 490
390 LOCATE 32,Y1:PRINT " ":T1=0:RETURN
400 IF Y2=15 AND X=0 THEN LOCATE 25,5:PRINT"ﾐｷﾞ ﾉ ｶﾁ!":GOTO 490
410 IF Y2=15 AND X=1 THEN LOCATE 25,5:PRINT"ﾐｷﾞ ﾉ ｶﾁ!":GOTO 490
420 IF Y2=16 AND X=1 THEN LOCATE 25,5:PRINT"ﾐｷﾞ ﾉ ｶﾁ!":GOTO 490
430 IF Y2=16 AND X=2 THEN LOCATE 25,5:PRINT"ﾐｷﾞ ﾉ ｶﾁ!":GOTO 490
440 IF Y2=16 AND X=0 THEN LOCATE 25,5:PRINT"ﾐｷﾞ ﾉ ｶﾁ!":GOTO 490
450 IF Y2=17 AND X=1 THEN LOCATE 25,5:PRINT"ﾐｷﾞ ﾉ ｶﾁ!":GOTO 490
460 IF Y2=17 AND X=2 THEN LOCATE 25,5:PRINT"ﾐｷﾞ ﾉ ｶﾁ!":GOTO 490
470 IF Y2=17 AND X=3 THEN LOCATE 25,5:PRINT"ﾐｷﾞ ﾉ ｶﾁ!":GOTO 490
480 LOCATE 7,Y2:PRINT " ":T2=0:RETURN
490 PAUSE 10:RUN
```

リスト2　GUN改.BAS

```
300 'ｺﾝﾅﾝ ﾃﾞﾏｼﾀｹﾄﾞｵｰ
310 LEFT=LEFT+(Y1=15 AND Y=4)+(Y1=15 AND Y=5)
320 LEFT=LEFT+(Y1=16 AND Y=4)+(Y1=16 AND Y=5)+(Y1=16 AND Y=6)
330 LEFT=LEFT+(Y1=17 AND Y=5)+(Y1=17 AND Y=6)+(Y1=17 AND Y=7)
340 IF LEFT THEN LOCATE 5,5:PRINT"ﾋﾀﾞﾘ ﾉ ｶﾁ!":GOTO 410
350 LOCATE 32,Y1:PRINT " ":T1=0:RETURN
360 RIGHT=RIGHT+(Y2=15 AND X=0)+(Y2=15 AND X=1)
370 RIGHT=RIGHT+(Y2=16 AND X=0)+(Y2=16 AND X=1)+(Y2=16 AND X=2)
380 RIGHT=RIGHT+(Y2=17 AND X=1)+(Y2=17 AND X=2)+(Y2=17 AND X=3)
390 IF RIGHT THEN LOCATE 25,5:PRINT"ﾐｷﾞ ﾉ ｶﾁ!":GOTO 410
400 LOCATE 7,Y2:PRINT " ":T2=0:RETURN
410 PAUSE 10:RUN
```

▶おおっ！　4年10カ月ぶりにパリパリの紙だ。　　伊藤　直也(24)静岡県

リスト3 PIC_MENU.BAS

```
10 /* pic_menu copyright 1994 t.sasaki
20 dim float g(65535)
30 dim str dn(255),dd(255),dt(255)
40 dim db(255)
50 int de,ms,me=5,cx,cy
60 str cd,wc
70 dim float li(1023)
80 screen 1,3,1,1
90 console ,,0
100 print "pic_menu"
110 print "input dir"
120 linput "";cd
130 if right$(cd,1)<>"¥" then cd=cd+"¥"
140 pic_menu()
150 home(0,0,0)
160 wipe()
170 end
180 /*-----------------------
190 func pic_menu()               /* 導入部
200   dim str ddd(0)
210   dim dddd(0)
220   str fn[63],da,tm,bt
230   int l,fo,dde
240   print "pic_menu icon check"
250   de=dir(cd+"*.pmi",dn,db,dd,dt)
260   print cd;"*.pmi=";
270   print using "###";de
280   if de>0 then {
290     for l=0 to de-1
300       fn=cd+left$(dn(l),len(dn(l))-3)+"pic"
310       if dir(fn,ddd,dddd,ddd,ddd)<>1 then {
320         print "delete ";cd+dn(l)
330         fdelete(cd+dn(l))
340       }
350     next
360   }
370   de=dir(cd+"*.pic",dn,db,dd,dt)
380   print cd;"*.pic=";
390   print using "###";de
400   if de<1 then end
410   for l=0 to de-1
420     dn(l)=left$(dn(l),len(dn(l))-4)
430     if dir(cd+dn(l)+".pmi",ddd,dddd,ddd,ddd)<>1 then {
440       pmi_save(l)
450     } else {
460       fo=fopen(cd+dn(l)+".pmi","r")
470       freads(bt,fo)
480       freads(da,fo)
490       freads(tm,fo)
500       if bt<>itoa(db(l)) or da<>dd(l) or tm<>dt(l) then {
510         pmi_save(l)
520       }
530       fclose(fo)
540     }
550   next
560   pmi_load_shell()
570 endfunc
580 /*-----------------------
590 func pmi_load(no)             /* アイコン読み込み
600   int y,l,ro,ls,po,spo
610   dim fo(6)
620   if no<0 or no>(de-1)¥7 then return()
630   if ms>no then ro=2 :ms=no :me=ms+5
640   if me<no then ro=1 :me=no :ms=me-5
650   y=(no mod 6)*85
660   if no*7+6>=de then ls=(de-1) mod 7 else ls=6
670   if ro=1 or ro=2 then {
680     if ro=1 then po=-8192 else po=-128
690     for l=0 to ls
700       fo(l)=fopen(cd+dn(no*7+l)+".pmi","r")
710       fseek(fo(l),po,2)
720     next
730   }
740   if ro=1 then {
750     for l=1 to 64
760       home(0,0,y+l)
770       line(0,y+l,511,y+l,0)
780       for ll=0 to ls
790         fread(li,16,fo(ll))
800         put(ll*72+8,y+l,ll*72+63,y+l,li)
810       next
820     next
830     fcloseall()
840     for l=65 to 85
850       home(0,0,y+l)
860     next
870     fill(0,y+65,511,y+65+12,0)
880     for l=0 to 6
890 symbol(l*72+8,y+65,left$(dn(no*7+l),8),1,1,0,65535,0)
900     next
910     return()
920   }
930   if ro=2 then {
940     for l=0 to 20
950       home(0,0,y+85-l)
960     next
970     fill(0,y+65,511,y+65+12,0)
980     for l=0 to 6
990 symbol(l*72+8,y+65,left$(dn(no*7+l),8),1,1,0,65535,0)
1000    next
1010    spo=-256
1020    for l=21 to 84
1030      if l=84 then spo=0
1040      home(0,0,y+85-l)
1050      line(0,y+85-l,511,y+85-l,0)
1060      for ll=0 to ls
1070        fread(li,16,fo(ll))
1080        put(ll*72+8,y+85-l,ll*72+63,y+85-l,li)
1090        fseek(fo(ll),spo,1)
1100      next
1110    next
1120    fcloseall()
1130    return()
1140  }
1150  for l=0 to ls
1160    fo(0)=fopen(cd+dn(no*7+l)+".pmi","r")
1170    fseek(fo(0),-8192,2)
1180    fread(li,1024,fo(0))
1190    fclose(fo(0))
1200    put(l*72+8,y+1,l*72+8+63,y+63+1,li)
1210 symbol(l*72+8,y+65,left$(dn(no*7+l),8),1,1,0,65535,0)
1220  next
1230 endfunc
1240 /*-----------------------
1250 func pmi_save(no)            /* アイコン書き込み
1260  int l,fo
1270  locate 0,csrlin
1280  print using " ###";no+1;
1290  apic_load(cd+dn(no)+".pic",0,0)
1300  for l=0 to 63
1310    get(l*8,0,l*8,511,li)
1320    put(l  ,0,l  ,511,li)
1330  next
1340  for l=0 to 63
1350    get(0,l*8,63,l*8,li)
1360    put(0,l  ,63,l  ,li)
1370  next
1380  get(0,0,63,63,li)
1390  fo=fopen(cd+dn(no)+".pmi","c")
1400  fwrites(itoa(db(no))+chr$(13)+chr$(10),fo)
1410  fwrites(dd(no)+chr$(13)+chr$(10),fo)
1420  fwrites(dt(no)+chr$(13)+chr$(10),fo)
1430  fwrite(li,1024,fo)
1440  fclose(fo)
1450  wipe()
1460 endfunc
1470 /*-----------------------
1480 func pmi_load_shell()        /* ファイルセレクト部分
1490  int l,ls,ik
1500  cls
1510  if de¥7>4 then ls=5 else ls=de¥7
1520  for l=0 to ls
1530    pmi_load(l)
1540  next
1550  sps()
1560  repeat
1570    sp_move(32,cx*72+8 ,cy*85+79,1)
1580    sp_move(33,cx*72+16+8,cy*85+79,1)
1590    sp_move(34,cx*72+32+8,cy*85+79,1)
1600    sp_move(35,cx*72+48+8,cy*85+79,1)
1610    repeat
1620      ik=asc(inkey$(0))
1630    until ik>0
1640    cx=cx-pow(ik=29 and cx>0 ,2)+pow(ik=28 and cx<6,2)
1650    cy=cy-pow(ik=30 and cy>-1,2)+pow(ik=31 and cy<6,2)
1660    if cy=6 or cy=-1 then {
1670      sp_off(32,35)
1680      if cy=6  then pmi_load(me+1) :cy=5
1690      if cy=-1 then pmi_load(ms-1) :cy=0
1700      sp_on(32,35)
1710    }
1720    if ik=13 then pic_load(ms,cy,cx)
1730  until ik=27
1740  sp_off()
1750 endfunc
1760 /*-----------------------
1770 func sps()                   /* スプライトセット
1780  dim char sp(255)
1790  int l
1800  sp_init()
1810  sp_clr()
1820  sp_disp(1)
1830  sp_on()
1840  sp_color(1,1)
1850  sp_color(2,65535)
1860  for l=0 to 15
1870    sp(l)=2
1880    sp(l+16)=2
1890  next
1900  sp_def(1,sp,1)
1910  for l=0 to 15
1920    sp(l)=1
1930    sp(l+16)=0
1940  next
1950  sp_def(0,sp,1)
1960  for l=0 to 31
1970    sp_move(l,l*16,0,0)
1980  next
1990 endfunc
2000 /*-----------------------
2010 func pic_load(a,b,c)         /* 選ばれたＰＩＣを表示
2020  int ik,hy,p
2030  str fn
2040  if de<=a*7+b*7+c then return()
2050  fn=cd+dn(a*7+b*7+c)+".pic"
2060  hy=(a*85) mod 512
2070  get(0,0,511,511,g)
2080  wipe()
2090  sp_off()
2100  home(0,0,0)
2110  apic_load(fn,0,0)
2120  repeat
2130    ik=asc(inkey$(0))
2140  until ik>0
2150  if ik=27 then end
2160  wipe()
2170  home(0,0,hy)
2180  put(0,0,511,511,g)
2190  sp_on()
2200 endfunc
2210 /*-----------------------
```

▶メガディスプレイ最高ッス。1024×714ドット(CZ-614D)で幸せいっぱい。でも，おかげで31kHzは画面外まで縦長～。　木村　直也(22)東京都

リスト4 B_BALL.C

```
  1: #include  "basic.h"
  2: #include  "basic0.h"
  3: #include  "music.h"
  4: #include  "stdlib.h"
  5: #include  "math.h"
  6: #include  "stick.h"
  7: #include  "graph.h"
  8: #include  "sprite.h"
  9: #define   E_MAX  500
 10: #define   UP     2
 11: #define   DOWN   4
 12: #define   AT     4
 13: static    int x[1+1],y[1+1],en[1+1],gx,gy,bg,wx,wy;
 14: static    unsigned char ch[256],s[1+1],m[1+1],st,tr;
 15: static    int i,j,lose;
 16: void PLAYER();  void VERO();    void DISP_CHARA();
 17: void DEATH();   void MUSIC_SET();
 18: void B_WAIT();  void SCROLL();  void ENDING();
 19: void SP_MAKE(); void PAT_SET(); void KAITEN();
 20: void main()
 21: {
 22:   double d;MUSIC_SET();
 23:   m_play('NASI','NASI','NASI','NASI','NASI','NASI',7,8);
 24:   SP_MAKE();
 25:   screen( 0,1,1,1);window(0,0,511,511);
 26:   console('NASI','NASI',0);b_csw(0);
 27:   sp_disp(1);palet(1,2048*10+64*10+2*10+1);
 28:   for(i=0;i<4;i++){
 29:     palet(7+i,64*(6+i*8)+2*i*8);
 30:     palet(11+i,2*(6+i*8)+64*i*8);}
 31:   apage(2);
 32:   for(i=0;i<4096;i++){
 33:     d=rnd()*8;
 34:     pset((int)((i%64)*8+d),(int)((i/64)*8+d),i%15);}
 35:   apage(0);
 36:   symbol(0,20,"BATTLE BALL",2,4,0,6,0);
 37:   symbol(64,128,"START -- push 1",1,1,1,15,0);
 38:   symbol(64,160,"E N D -- push 2",1,1,1,15,0);
 39:   for(i=0;i<2;i++){
 40:     fill(128*i+9,19+256,128*i+114,41+256,15);}
 41:   while( lose!=10 ){
 42:     vpage(5);apage(1);wipe();
 43:     circle(255,255,224,1,0,360,256);paint(255,255,1);
 44:     vpage(5);
 45:     m_play('NASI','NASI','NASI','NASI','NASI','NASI',7,8);
 46:     x[0]=144;x[1]=368;y[0]=255;y[1]=255;en[0]=0;en[1]=0;
 47:     DISP_CHARA(0);DISP_CHARA(1);sp_on(0,127);
 48:     home(0,0,0);B_WAIT();home(1,128,128);
 49:     m_play('NASI','NASI',3,4,'NASI','NASI','NASI','NASI');
 50:     ( tr==2 )?(lose=10):(lose=0);
 51:     vpage(15);apage(0);fill(0,316,255,511,0);
 52:     m_play('NASI','NASI','NASI','NASI','NASI','NASI',7,8);
 53:     for(i=0;i<2561;i++){home(0,0,i/10);    }
 54:     while( lose==0 ) {PLAYER(0);PLAYER(1);SCROLL();}
 55:     apage(0);
 56:     m_stop('NASI','NASI','NASI','NASI','NASI','NASI','NASI'
8);
 57:     switch(lose){
 58:       case 1:
 59:         for(i=0;i<=6000;i++){sp_off(i/2000,'NASI');}
 60:         symbol(20,120+256,"WINNER 2P",2,6,0,14,0);
 61:         break;
 62:       case 2:
 63:         for(i=0;i<=6000;i++){sp_off(i/2000+4,'NASI');}
 64:         symbol(20,120+256,"WINNER 1P",2,6,0,10,0);
 65:         break;
 66:       case 10:
 67:         symbol(20,120+256," END END ",2,6,0,15,0);
 68:         break;
 69:       default:
 70:         for(i=0;i<=6000;i++){
 71:           sp_off(i/2000,'NASI');sp_off(i/2000+4,'NASI');}
 72:         symbol(20,120+256,"DRAW GAME",2,6,0,6,0);
 73:         break;
 74:     }
 75:     vpage(5);sp_off(0,127);B_WAIT();
 76:   }
 77:   m_stop(1,2,3,4,5,6,7,8);
 78:   b_exit(0);
 79: }
 80: void  PLAYER(p)
 81: unsigned char p;
 82: {
 83:   unsigned char e;
 84:   double lx,ly;
 85:   int vx,vy,ox,oy;
 86:   ox= x[p];oy= y[p];e=( p==0 );
 87:   st= stick(p+1);tr= strig(p+1);VERO(m[p],&vx,&vy);
 88:   if( tr==1 & s[p]!=1 ){
 89:     m_play(1,2,'NASI','NASI','NASI','NASI','NASI','NASI');
 90:   }
 91:   if( tr!=2 & st!=0 & en[p]>(E_MAX/4) ){
 92:     x[p]+=vx*(1+AT*( tr==1 ));y[p]+=vy*(1+AT*( tr==1 ));
 93:   }
 94:   if( ( abs(x[0]-x[1])>240 ) ){x[p]= ox;};
 95:   if( ( abs(y[0]-y[1])>240 ) ){y[p]= oy;}
 96:   apage(1);
 97:   if( st!=0 & tr==2 & en[p]>(E_MAX*2/5) & s[p]!=2 ){
 98:     m_play('NASI','NASI','NASI','NASI',5,6,'NASI','NASI');
 99:     wx=x[p]+vx*32;wy=y[p]+vy*32;en[p]-=(E_MAX/2);
100:     circle(wx,wy,16,0,0,360,256);paint(wx,wy,0);
101:     if ( ( s[e]==1 )*( en[e]>E_MAX*3/4 )!=1 ) DEATH(e);
102:   }
103:   lx=abs(x[0]-x[1]);ly=abs(y[0]-y[1]);
104:   if ( lx<30 & ly<30 ){
105:     if ( (int)sqrt(lx*lx+ly*ly)<27 ){
106:       if ( tr!=1 ){ x[p]=ox;y[p]=oy; }
107:       if ( tr==1 & s[e]==1 ){
108:         m_play('NASI','NASI',3,4,'NASI','NASI','NASI','NASI
');
109:         x[p]= x[p]-vx*16;y[p]= y[p]-vy*16;en[p]=0;
110:         VERO(m[e],&vx,&vy);
111:         x[e]= x[e]-vx*16;y[e]= y[e]-vy*16;en[e]=0;
112:       }
113:       if ( tr==1 & s[e]!=1 ){
114:         m_play('NASI','NASI',3,4,'NASI','NASI','NASI','NASI
');
115:         for( j=0;j<32;j++){
116:           x[p]+=vx;y[p]+=vy;x[e]+=vx*AT/2;y[e]+=vy*AT/2;
117:           DISP_CHARA(p);DISP_CHARA(e);SCROLL();
118:           DEATH(e);if( lose!=0 ) break;
119:         }
120:       }
121:     }
122:   }
123:   en[p]+=UP*( tr==0 )*( en[p]<E_MAX );
124:   en[p]-=(DOWN*( tr==1 ))*( st!=0 );
125:   en[p]*=( en[p]>0 );s[p]=tr;m[p]=st;
126:   apage(0);
127:   fill(128*p+10+(en[p]*50/E_MAX)*2,20+256,
128:        128*p+10+(en[p]*50/E_MAX)*2,40+256,
129:        7+p*4+( en[p]>(E_MAX/4) )+( en[p]>(E_MAX*2/5) )+( en
[p]>(E_MAX*3/4) ));
130:   if ( en[p]<E_MAX ){
131:     fill(128*p+12+en[p]*100/E_MAX,
132:          20+256,128*p+110,40+256,0);
133:   }
134:   if ( ( tr==1 )*( en[p]>E_MAX*3/4 )!=1 ) DEATH(p);
135:   DISP_CHARA(p);
136: }
137: void VERO(s,vx,vy)
138: char s;int *vx,*vy;
139: {
140:   *vx= ((s-1)%3)-1;*vy= -(((s-1)/3)-1);
141: }
142: void DISP_CHARA(c)
143: unsigned char c;
144: {
145:   gx= ((x[0]+x[1])/2-127);
146:   gx= gx*( gx>=0 )*( gx<256 )+256*( gx>=256 );
147:   gy= ((y[0]+y[1])/2-127);
148:   gy= gy*( gy>=0 )*( gy<256 )+256*( gy>=256 );
149:   home(1,gx,gy);wx=x[c]-gx;wy=y[c]-gy;
150:   for(i=0 ;i<= 3;i++){
151:     sp_set(c*4+i,wx+14*(i % 2)+1,wy+14*(i/2)+1,
152:     256*(c+1)+st*( tr==1 )*( en[c]>(E_MAX/4))+i*10,3);}
153: }
154: void DEATH(p)
155: unsigned char p;
156: {
157:   apage(1);
158:   lose+=( point(x[p],y[p])==0 )*( lose!=(p+1) )*(p+1);
159: }
160: void B_WAIT()
161: {
162:   do {tr=strig(1);} while( tr!=0 );
163:   do {tr=strig(1);SCROLL();for(i=0;i<1000;i++){}} while( tr
==0 );
164: }
165: void SCROLL()
166: {
167:   bg=(bg+1)*(bg<512);home(2,bg,bg);
168: }
169: void MUSIC_SET()
170: {
171:    m_init();
172:    for(i=1;i<=8;i++){m_alloc(i,100);m_assign(i,(i+1)/2);}
173:    m_trk(1,"o2 l16 v11 @68 g&f&e");
174:    m_trk(2,"o1 l64 v14 @32 gg&g&ga");
175:    m_trk(3,"o1 l64 v11 @ 6 g&a&b");
176:    m_trk(4,"o1 l16 v12 @38 |:255 g8gg gfga gbga f8fg fgfb f
bfg ffee d8ff:|");
177: }
178: void SP_MAKE()
179: {
180:   double rad;screen(0,2,1,1);sp_init();
181:   for(i=0;i<5+1;i++){
182:     sp_color(10+i,(i+1)*64*5,1);
183:     sp_color(10+i,(i+1)*2*5,2);}
184:   for(i=0;i<6;i++){
185:     circle(15+i,15,15-i*2,10+i,0,360,192);
186:     paint(15+i,15,10+i);
187:     circle(15,63,15-i*2,10+i,0,360,256);
188:     paint(15,63,10+i);}
189:   PAT_SET(0);
190:   for(i=0;i<3;i++){
191:     rad=(5+i)*pi()/4;KAITEN(rad);PAT_SET(i+1);}
192:   for(i=0;i<2;i++){
193:     rad=(1-i)*pi();KAITEN(rad);PAT_SET(i*2+4);}
194:   for(i=0;i<3;i++){
195:     rad=(3-i)*pi()/4;KAITEN(rad);PAT_SET(i+7);}
196: }
197: void KAITEN(rad,n)
198: double rad;
199: {
200:   char col;
201:   double x,y,c,s;c=cos(rad);s=sin(rad);
202:   for(j=0;j<1024;j++){x=(j%32)-15;y=(j/32)-15;
203:     col=point((int)(x*c-y*s)+15,(int)(x*s+y*c)+15);
204:     pset((int)(15+x),(int)(63+y),col);}
205: }
206: void PAT_SET(n)
207: char n;
208: {
209:   for(j=0;j<4;j++){
210:     get(0+(j%2)*16,48+(j/2)*16,15+(j%2)*16,63+(j/2)*16,&ch,
256);
211:     sp_def(n+j*10,&ch,1);}
212: }
```

NEW PRODUCTS

新製品紹介

X680x0 Develop. & libc II

Nakamori Akira 中森 章

ついに，というか当然というかX68k Programming Seriesの第1弾のDevelop.と第2弾のlibcの追補版が発売されました。X680x0 Develop. & libc IIです。これはX680x0マシン完全対応を目的とした改訂です。コンパイラが68030コードを生成できなかったXCのNEW KITのときとは異なり，X68030で68030コードでのコンパイル，アセンブル，デバッグを可能にします。

また，第1弾のDevelop.のときは，別途XCのライブラリか第2弾であるlibcがないと役に立ちませんでしたが，今回はライブラリであるlibcがセットになっています。これでXCを持っていない人でもCコンパイラでのプログラム開発が可能になりました。なんと挑戦的な企画でしょうか。

▶ ディスクの中身 ◀

付属ディスクは2枚です。インストールのためにはディスク1の1枚で十分です。コンパイラ(Develop.)パッケージ，ライブラリ(libc)パッケージ，ライブラリ(libc)のソースパッケージがTAR+GZIP形式で圧縮されて格納されています。それと，解凍用のTXF.Rとライブラリのソースを解凍するときに必要になるTwentyOne.Xとインストールプログラムからなります。

ディスク2にはパソコン通信などで配布されている各プログラムが配布されているそのままの形式で収録されています。

●GCCの主な変更点

GCCの最大の変更点として挙げられるのは対象CPUの拡張です。X68030用の68030コードだけではなく68040のコードも生成できるようになっています。ただし，シャープ純正のアセンブラではアセンブルできないコードを生成することもあるようです。純正のアセンブラも68030だけでなく68040のコードをアセンブルできるようになっていただけにちょっと惜しい気もします(その点はHASで十分にカバーされているのでなにも問題はありませんが)。

表 章構成

Chapter0	インストール
Chapter1	X680x0 GCC
Chapter2	X680x0 HAS
Chapter3	X680x0 HLK
Chapter4	X680x0 GDB
Chapter5	Develop.便利帳
Chapter6	最新版の概要 (libc)
Chapter7	libc便利帳
Chapter8	Appendix A
Chapter9	Appendix B

それ以外の主な拡張は予約語の追加です。割り込み関数宣言の予約語，プログラムカウンタ間接アドレッシングをコンパイラに強制する予約語，それにMS-DOS上のコンパイラ用に作成されたソースを簡便にコンパイルするための予約語です。

●HASの主な変更点

HASの最大の変更点もサポートするCPUを追加したことです。従来は68000と68010だけでしたが，68020,68030,68040がサポートされています。また，浮動小数点コプロセッサ68881,68882はもちろんのこと，68020専用のメモリ管理ユニット68851までもサポートしているのは驚きです。

また，ソースファイルの中で浮動小数点の実数表記が可能になっています。

そのほか，より小さなオブジェクトを出力するための最適化機能の強化，ローカルラベルの拡張などがあります。

●HLKの主な変更点

リンカ自身はCPUの依存度が低いため，大幅な変更はありません。アラインメントの値を2以外に変更できるようになった以外はバグフィックスが主なところです。

●GDBの主な変更点

最大の変更点は，これもCPUの追加です。ただし，X68000とX68030だけを対象としているようで，68000,68030,68881,68882のコードのみの対応のようです。

あとはスーパーバイザモードで動作するプログラムのデバッグが可能になった点，libcをリンクしたプログラムのデバッグに対応した点などです。

●libcの主な変更点

バグフィックスとともにいくつかの関数の仕様が変更になったり，追加されたりしています。

▶ マニュアルの概要 ◀

マニュアルの章立ては表のようになっています。このマニュアルでは上述した変更点が詳しく解説されています。ただし，これだけではX680x0 Develop. & libcのマニュアルとして完全なものではありません。あくまでも，第1弾のDevelop.と第2弾のlibcのマニュアルを持っているものとして話が進められています。今回，Develop.とlibcのマニュアル版が同時に発売されていますが，これは，いきなりX680x0 Develop. & libc IIを購入する人への配慮なのでしょう。

なお，Chapter8のAppendix Aには今回大幅に拡張されたアセンブリ言語の表記法が，Chapter9のAppendix Bには各ツールのオプションスイッチと診断メッセージ，アセンブラ疑似命令一覧，GDBコマンド一覧，ライブラリ関数機能別一覧が掲載されています。

*　　　*　　　*

今回のX680x0 Develop. & libc IIでXCの立場がなくなってしまったというのが素直な感想です。ただ，libcにはX-BASICのライブラリはあるものの，サポートしているのがAD PCM関係の関数のみなのでX-BASICの関数を使うときはXCに頼ることとなり，X-BASICコンパイラとしての役目はまだ終わっていません。また，libcのソースはC言語で記述してあるので，アセンブリ言語でライブラリを記述してあるXCに比べて生成されるコードのバイト数が大きくなるという欠点を持っています。そういうわけでXCの生き延びる道は残されていますが……。筆者個人では生成されるコードの大きさが気になるのでライブラリだけはXCのものを使おうと思っています。

X680x0 Develop. & libc II　2,900円(税込)
ソフトバンク　☎03(5642)8101

SIDE A

予定調和と決別する

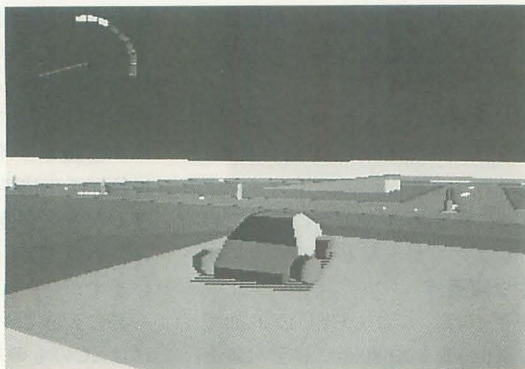
Tan Akihiko 丹 明彦

シミュレーションモデルを見直し，繰り返し実験を行っていく
まだまだ理解の及ばないところもあるが，やらないことにははじまらない
謎の暴走に頭を悩ませつつ，今回も車の挙動を制御するため突き進んでいく

今回も引き続き，車のコーナリングを扱う。車の挙動としては前回とあまり変わらないが，予定調和的な要素を排除したことによって，より先につながるシミュレーションモデルになったと考えている。

なにが予定調和だったのか

前回取り扱ったシミュレーションモデルは,「コーナリング＝定常円旋回」を前提としていた。つまり，ステアリングを切ることによって発生したコーナリングフォース(F)と，車の質量(m)と，車の速度(v)を，円運動の公式「$F=mv^2/r$」に代入してrについて解くことによって旋回半径（r）と予想される軌跡を求め，車体を移動および回転させていた。

これのどこが予定調和だったかというと,「ステアリングを切れば円旋回運動になる」という結果を予想してシミュレーションモデルを立てていた，その姿勢が予定調和的だったのである。いまのうちはまだそれでいいのだが，じきに4輪ドリフトを取り扱うようにしなくてはならなくなる。そのときに「ステアリング→円旋回モード」という構図だけでは対処しきれなくなる。そこで「ドリフトモード」などを次々と導入してプログラムをif文の塊にしてしまうのであろう。モードを山ほど設けて実車の挙動に近づけるというアプローチを否定するわけではない。ただ，車の挙動は複数の現象が同時に発生しながら複雑に決まっていくので，モードで押していくと現象の数が爆発しかねないのである。

タコメーターが表示され，ちょっとは進歩したモデル

ここは徹底的に現象の力学計算だけで押し通し，望む車の挙動を発生させてみたい。今回の場合でいえば，タイヤ周辺に発生する力だけで瞬間瞬間の車の姿勢を求めて，ステアリングを切れば車の挙動が「結果的に」円旋回運動になるようなシミュレーションモデルを構築したいということなのだ。

回転に関する運動方程式

コーナリングを力学的に記述するうえで避けて通れないのが，この回転に関する運動方程式である。通常，車は剛体と考えてよい。「剛体」とは，簡単にいえば，大きさをもった物体であり，大きさをもたない「質点」よりも扱いが若干複雑になる。質点の運動は平行移動とその速度だけ考えればよい（旋回運動にしても，瞬間瞬間を捉えれば平行移動運動である）が，剛体の運動は自身の回転運動とその角速度をも考える必要が出てくる。逆にいえば，質点とはこの回転運動を無視する目的で考えられた物理学上の概念ということもできる。

・平行移動に関する運動方程式

平行移動運動を支配する方程式は，ご存じ，

$$F=ma$$
$$a=dv/dt$$

である(図1)。Fは力，mは質量，aは加速度，vは速度，tは時刻である。dv/dtはvをtで微分したものである(単位時間あたりの変化量)。Fとaとvはベクトルで表示することもできる（している)。

使い方は，質量mが与えられている物体に力Fを加えると加速度aが発生し，速度がa×dt増加する。

・回転に関する運動方程式

回転運動を支配する方程式は，

$\tau = I(d\omega/dt)$

である（図2）。τはトルク，Ｉは慣性モーメント，ωは角速度である。

剛体に回転運動を発生させるのはトルクτである。トルクとは，力点に加えた力の大きさと重心から力点までの距離をかけたものである。同じ大きさの力でも，重心から離れた場所に力をかけたほうが大きなトルクを与えることになり，より高速に物体を回転させることができる。いわゆる「てこの原理」というやつである。同じ物体でも端をもったほうが回しやすいのは，こういう理屈なのである。

慣性モーメントＩは物体の回りやすさを示す量である。この値が小さいほど物体は回しやすい。同じ物体でも，回転軸のとり方によって慣性モーメントは異なる（図3）。角速度ωは，単位時間あたりの角度の変化量である。さらにその単位時間あたりの変化量（dω/dt）は角加速度である。

ステアリングを切ったときの車の運動

剛体の運動について知識を得たところで，車の運動を記述する。といっても大部分は前回と同様の計算である。異なるところは，旋回に関する部分である。旋回運動は，前回は「旋回中心に対する回転運動」であったのが，「向心力による平行移動」と「トルクによる車体自体の回転」の２つの運動によって表現される（図4）。これに伴い，旋回半径と旋回中心に関する部分を削除した。

向心力による平行移動は次のようになる。まずタイヤから発生する力の，進行方向と垂直な成分を向心力とみなす。次に，角速度から生じる遠心力を求める。最後に，これらの差を車体の質量で割ったものを加速度として用いる。トルクによる車体自体の回転は次のようになる。まずトルクから角加速度を求め，dtをかけて角速度に加える。そして角速度に従って車体を回転させる。

コーナリングの不安定性に関する考察

幸い，今回も車が一応安定した挙動を示しているのだが，質量やコーナリングフォース，それに慣性モーメントなどの諸定数の値を変えると，簡単に妙な挙動をしたり暴走したりしてしまう。たとえば一定の方向に旋回を続けると角速度がつきすぎてスピン状態に陥ってしまう。これはどうやら，数値的な不安定性によるものではなさそうである。

トルクは車体を回転させるための力として導入した。しかし，このトルクの発生源は，あくまでタイヤのコーナリングフォースである。これは私の推測だが，タイヤから発生したトルクが車体をタイヤの舵角以上に回転させているらしい。物理的には変なのだが，数値シミュレーションの悲しさ，どんなに変な現象でも計算だけはできてしまうのだ。

じつはこのへんの事情は制動力でも同じである。制動力はあくまで速度を殺すための力であり，きちんと働けば車は停止するのである。現時点では，制動力は進行方向と逆向きの力として計算されている。たとえば前進しているときに制動力が強すぎると，勢い余って後ろに走り出してしまうのだ。これは早急に解決すべき問題といえる。

駆動力についても，現在のようないい加減な値でなく，きちんとトルクカーブを導入する時期が近づいている。その前に，４輪独立して路面を捉え，荷重移動やグリップの度合いを計算することが必要だ。

図1　平行移動

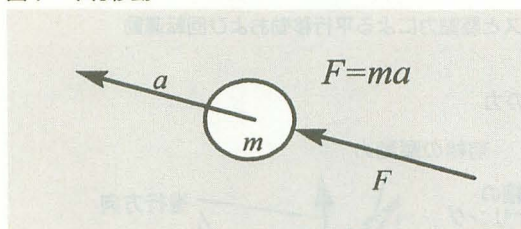



図2　回転

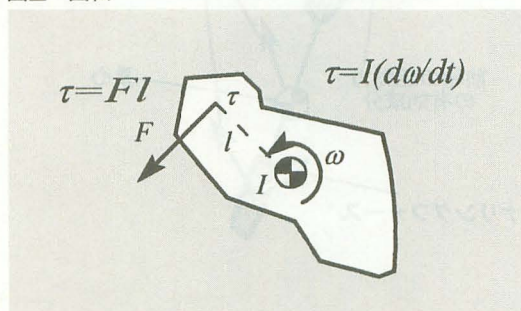
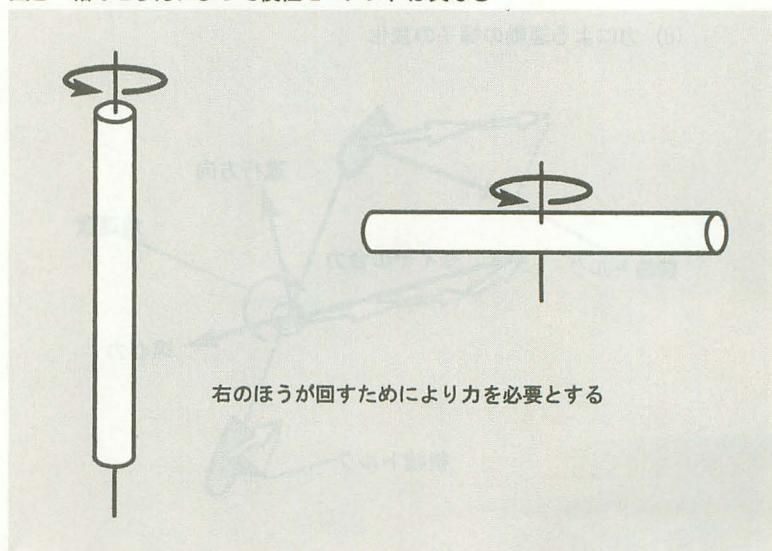
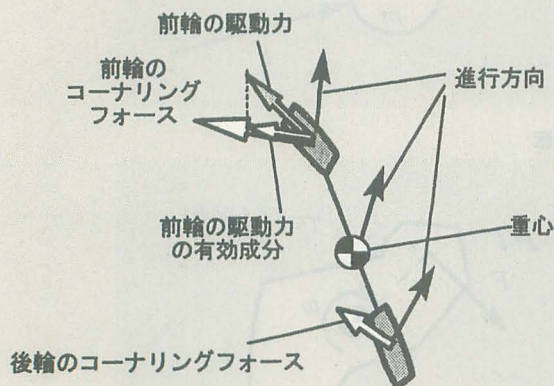
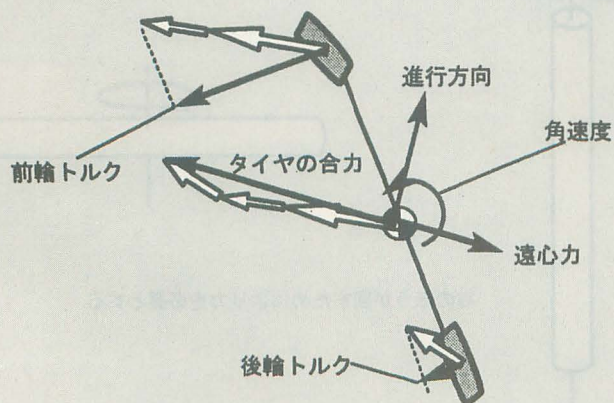



図3　軸のとり方によって慣性モーメントは異なる

そうすれば，ホイールスピンのような現象も力学的に実現でき，アクセルターンなどが可能になっていくことだろう。

教科書と現実のギャップについて

自動車の挙動について深く知ろうと思えば，自動車力学の本を教科書として読むことは不可欠である。たとえば自動車のコーナリングのメカニズムひとつとっても，タイヤが向いた方向に転がっていくというほど単純なものではないことがおわかりいただけたことと思う。

しかし，自動車力学の本を読めばドライビングシミュレーションプログラムが簡単に書けるかといえば，そんなことはないのも事実である。

図4 コーナリングフォースと駆動力による平行移動および回転運動

理由のひとつは，微分方程式で記述されている運動方程式を，リアルタイムシミュレーションの場に持ち込むための手法。まあこれは大した問題ではない。1フレームあたりの表示時間をdtとみなして近似的に積分してしまえばいい。微分方程式の解き方としては単純だ。まだdtが大きな場合は安定させにくいという問題があるが（いわゆる陽解法で解いているため，dtが大きいと解が発散する)。しかし，結果オーライということで，大きな問題にはなっていないようである。

さらに，なんといっても自動車力学の本で扱われている個々の現象が，基本的に単体でしか分析されていないうえに，ほとんどが定常状態を基本として書かれていることにある。もちろん，数式を使って解析的に記述しようとすれば，そうせざるをえないのだが，現実の車の挙動，特にレーシングカーの挙動は複数の現象，それも過渡現象の塊なのである。たとえばコーナリングにしても，教科書ではコーナリングフォースの発生は「ある程度の速い速度で前方に走っている」状態を想定しているが，いざ実際のシミュレータを書く段になると，極低速や後退，さらには横走りという状態にも対応できるように書かなくてはならない。その間も絶えず外乱要因はあるわけだし，すべての要素を同時に計算している現状ではどんな怪現象が起きても不思議ではない。事実，デバッグ途中は謎の暴走（プログラムの暴走ではない，文字どおりの暴走）やスピンにも悩まされている。まだコーナリングしか扱っていないこの段階でである。

いまのところ，教科書から使えそうな記述を抜き出しつつ，ある程度の不正確さを覚悟のうえで適用しているのが現状である。破綻しないことを心の底から祈りつつ，今回はここまで。

力学シミュレーションに関する補足事項

前回，力の単位として「N」を基本単位として使い，「kgf」は派生的に用いていたが，実は自動車工学の世界では，力は「kgf」を基本単位としている。したがって私のアプローチは，少なくとも自動車シミュレーションを行う人たちの間では異端ということになるらしい。

学問の世界ではこういう作法についてけっこう流派があるものである。確かに高校物理レベルの知識だけで無の状態から理解しようとするのであれば，MKS単位系で統一するのが美しくはある。しかし，実践経験に根ざした実用的単位系で自動車力学の体系が構成されている以上，逆らうのは無意味である。

とはいうものの，こういう事実は頭の中に入れておくだけにして，プログラムはとりあえずMKS単位系で書き続けていこうと考えている。

なお，本文での解説は単位系によらない書き方になっているので，単位系を気にするのはプログラムを記述するときだけである。

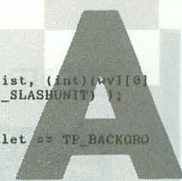
■リスト1 TDRIVE.C (参考)

```
  1: /*
  2:  *        tdrive.h
  3:  *        - 車の動作(テキスト画面)
  4:  *          Jul. 1994 - Aug. 1994  丹 明彦(Oh!X)
  5:  */
  6:
  7: #ifndef __TDRIVE_H__
  8: #define __TDRIVE_H__
  9:
 10: #include        <slash3/slmath.h>
 11:
 12: #define N_POINT         800
 13: #define N_OBJECT        4
 14:
 15: /* 単位変換(試作) */
 16: #define HOUR    *3600.0         /* 時間 */
 17: #define KM      *1000.0         /* キロメートル */
 18: #define KMPH    *(1.0 KM)/(1.0 HOUR)/* キロメートル毎時 */
 19: #define _KMPH   *(1.0 HOUR)/(1.0 KM)/* キロメートル毎時にする */
 20: #define G       9.8             /* 重力加速度 */
 21: #define KGF     *G              /* キログラム重 */
 22: #define DEG     *(M_PI/180.0)   /* 度 */
 23: #define _DEG    *(180.0/M_PI)   /* 度にする */
 24:
 25: /* SLASHとの長さ変換 */
 26: #define SLASHUNIT       *0.01   /* SLASH単位 */
 27: #define _SLASHUNIT      *100.0  /* SLASH単位にする */
 28:
 29: typedef struct _CarSpec {
 30:   double   width;     /* ボディ全幅 */
 31:   double   length;    /* ボディ全長 */
 32:   double   height;    /* ボディ全高 */
 33:   double   fshaft;    /* Fホイール幅 */
 34:   double   rshaft;    /* Rホイール幅 */
 35:   double   fwidth;    /* Fタイヤ幅 */
 36:   double   fradius;   /* Fタイヤ半径 */
 37:   double   rwidth;    /* Rタイヤ幅 */
 38:   double   rradius;   /* Rタイヤ半径 */
 39:   double   wheelbase; /* ホイールベース */
 40:   double   steeringratio;
 41:              /* ステアリング係数(マウスX座標と舵角の関係) */
 42:   int   frontdrive;   /* 前輪駆動 */
 43:   int   reardrive;    /* 後輪駆動 */
 44:   double    l1;       /* 前輪軸から重心までの距離 */
 45:   double    l2;       /* 後輪軸から重心までの距離 */
 46:   double   mass;      /* 質量(単位はkg) */
 47:   double   ibeta;     /* 慣性モーメント(単位はkgf.m/s^2) */
 48: } CarSpec;
 49:
 50: typedef struct _CarInfo {
 51:   VECTOR3   b;       /* 車体の位置 */
 52:   VECTOR3   ba;      /* 車体の基底ベクトル(α軸) */
 53:   VECTOR3   bb;      /* 車体の基底ベクトル(β軸) */
 54:   VECTOR3   bc;      /* 車体の基底ベクトル(γ軸) */
 55:   VECTOR3  dir;      /* 車体進行方向 */
 56:   VECTOR3  dirf;     /* タイヤ進行方向前輪 */
 57:   VECTOR3  dirr;     /* タイヤ進行方向後輪 */
 58:   VECTOR3   sidedir;/* 車体進行方向に向かって右方向 */
 59:   VECTOR3   vel;      /* 車体速度 */
 60:   VECTOR3   velf;     /* タイヤ速度前輪 */
 61:   VECTOR3   velr;     /* タイヤ速度後輪 */
 62:   VECTOR3   revf;     /* タイヤ回転方向前輪 */
 63:   VECTOR3   revr;     /* タイヤ回転方向後輪 */
 64:   int   revdf;        /* タイヤ回転方向前輪 */
 65:   int   revdr;        /* タイヤ回転方向後輪 */
 66:   double   scf;       /* スリップアングルの余弦前輪 */
 67:   double   scr;       /* スリップアングルの余弦後輪 */
 68:   double   cf;        /* コーナリングフォース前輪 */
 69:   double   cr;        /* コーナリングフォース後輪 */
 70:   VECTOR3   cff;      /* コーナリングフォースベクトル前輪 */
 71:   VECTOR3   cfr;      /* コーナリングフォースベクトル後輪 */
 72:   double  drvf;       /* 駆動力前輪 */
 73:   double  drvr;       /* 駆動力後輪 */
 74:   double   brkf;      /* 制動力前輪 */
 75:   double   brkr;      /* 制動力後輪 */
 76:   VECTOR3   trkf;     /* タイヤのトラクション前輪 */
 77:   VECTOR3   trkr;     /* タイヤのトラクション後輪 */
 78:   VECTOR3   tire;     /* タイヤの合力 */
 79:   VECTOR3   tiref;    /* タイヤの合力前輪 */
 80:   VECTOR3   tirer;    /* タイヤの合力後輪 */
 81:   double   force;     /* 力の大きさ */
 82:   double   ve;        /* 速度 */
 83:   double   vd;        /* 速度の車体方向成分 */
 84:   double   vs;        /* 速度の車体横方向成分 */
 85:   double   omega;     /* 角速度 */
 86:   int   rpm;          /* エンジン回転数 */
 87:   double   theta;     /* ステアリング角 */
 88:   double   phi_fr;    /* タイヤの転がり角(前右) */
 89:   double   phi_fl;    /* タイヤの転がり角(前左) */
 90:   double   phi_rr;    /* タイヤの転がり角(後右) */
 91:   double   phi_rl;    /* タイヤの転がり角(後左) */
 92:   double   stroke_fr; /* サスペンションのストローク(前右) */
 93:   double   stroke_fl; /* サスペンションのストローク(前左) */
 94:   double   stroke_rr; /* サスペンションのストローク(後右) */
 95:   double   stroke_rl; /* サスペンションのストローク(後左) */
 96:   double   height_fr; /* 接地点の高さ(前右) */
 97:   double   height_fl; /* 接地点の高さ(前左) */
 98:   double   height_rr; /* 接地点の高さ(後右) */
 99:   double   height_rl; /* 接地点の高さ(後左) */
100:   double   roll;        /* にせローリング */
101:   double   pitch;       /* にせピッチング */
102:   double   heightratio;   /* にせバウンド */
103: } CarInfo;
104:
105: typedef struct _Control {
106:   int   mousex;
107:   int   mousey;
108:   int   mouselb;
109:   int   mouserb;
110:   int   bskey;
111:   int   shiftkey;
112: } Control;
113:
114: #endif   /* __TDRIVE_H__ */
```

■リスト2 DRIVE.C (参考)

```
  1: /*
  2:  *        drive.c
  3:  *        - 車の動作
  4:  *          Jul. 1994 - Sep. 1994  丹 明彦(Oh!X)
  5:  */
  6:
  7: #define         __IOCS_INLINE__
  8: #include        <iocslib.h>
  9: #include        <stdio.h>
 10:
 11: #include        <slash3/slashlib.h>
 12: #include        <slash3/addprim.h>
 13: #include        <slash3/check.h>
 14: #include        <slash3/timedifference.h>
 15: #include        "slmath2.h"
 16: #include        <slash3/perfmon.h>
 17: #include        <slash3/doublebuffer.h>
 18:
 19: #include        "tdrive.h"
 20: #include        "textcolor.h"
 21:
 22: /* 最高速度(にせの駆動力を計算するため) */
 23: #define MAXSPEED                (200.0 KMPH)
 24: #define MAXSPEEDR               (20.0 KMPH)
 25:
 26: /* 旋回運動に関する定数または近似式 */
 27: #define CORNERINGFORCEF(C)      (20.0 KGF *C*acos(C) _DEG)
 28: #define CORNERINGFORCER(C)      (40.0 KGF *C*acos(C) _DEG)
 29: #define MASS                    (500.0)
 30: #define DRIVEFORCE              (400.0 KGF)
 31: #define BRAKEFORCEF             (500.0 KGF)
 32: #define BRAKEFORCER             (250.0 KGF)
 33:
 34: /* 慣性モーメント(単位はkgf.m/s^2) */
 35: #define IBETA                   (100.0 KGF)
 36:
 37: /* にせローリング/にせピッチング */
 38: #define ROLL            (3.0*M_PI/180)
 39: #define PITCH           (1.5*M_PI/180)
 40: #define HEIGHTRATIO     (0.0)
 41:
 42: int     ontime;         /* 時刻 */
 43: double  dt;             /* 時差 */
 44:
 45: /* 車の情報 */
 46: extern Control  control;
 47: extern int      phase;
 48:
 49: /* フィールドと衝突判定ポリゴンの実体はtcource.cで宣言されている */
 50: extern SLPOLYGONLIST    *collide_polygonlist;
 51: extern SLPOINTLIST      *collide_pointlist;
 52: /* フィールドの衝突判定情報の実体はtdrive.cで宣言されている */
 53: extern CHECKINFOLIST    *collide_checkinfolist;
 54:
 55: void    checkcollision( CarSpec *cs, CarInfo *ci )
 56: {
 57:   int  i, ry, rd;
 58:   double  l;
 59:   VECTOR3  v;
 60:
 61:   /* 検索起点 */
 62:   scalevector( v, -(cs->height)*0.5, ci[1-phase].bb );
 63:   addvector( v, v, ci[1-phase].b );
 64:   i = checks2( &ry, &rd, collide_checkinfolist, (int)(v[0] _SL
ASHUNIT), (int)(v[1] _SLASHUNIT), (int)(v[2] _SLASHUNIT) );
 65:   if ( i == -1 ) return;
 66:   /* 壁にぶつかったら前の状態に戻す */
 67:   if ( collide_polygonlist->polygon[i].palet == TP_BACKGROUND
) {
 68:     copyvector( ci[1-phase].b , ci[phase].b  );
 69:     copyvector( ci[1-phase].ba, ci[phase].ba );
 70:     copyvector( ci[1-phase].bb, ci[phase].bb );
 71:     copyvector( ci[1-phase].bc, ci[phase].bc );
 72:     return;
 73:   }
 74:   /* 新しい位置(y座標のみ変わる) */
 75:   ci[1-phase].b[1] = (double)ry SLASHUNIT;
 76:   /* 面法線を求めて新しいβ軸とする */
 77:   v[0] = -(double)collide_checkinfolist->ci[i].a;
 78:   v[1] = -(double)collide_checkinfolist->ci[i].b;
 79:   v[2] = -(double)collide_checkinfolist->ci[i].c;
 80:   normalizevector( ci[1-phase].bb, v );
 81:   /* 新しいγ軸(補正前のγ軸から新β軸成分を取り除いて正規化したもの) */
 82:   l = dotproduct( ci[1-phase].bb, ci[1-phase].bc );
 83:   scalevector( v, l, ci[1-phase].bb );
 84:   subvector( v, ci[1-phase].bc, v );
 85:   normalizevector( ci[1-phase].bc, v );
 86:   /* 新しいα軸(新β軸と新γ軸の外積) */
 87:   crossproduct( ci[1-phase].ba, ci[1-phase].bb, ci[1-phase].bc
 );
 88:   return;
 89: }
 90:
 91: void  initCar( CarSpec *cs, CarInfo *ci )
 92: {
 93:   cs->width   = 2.0;    /* ボディ全幅 */
 94:   cs->length  = 4.0;    /* ボディ全長 */
 95:   cs->height  = 1.2;    /* ボディ全高 */
```

```
 96:    cs->fshaft  = 2.3;     /* Fホイール幅 */
 97:    cs->rshaft  = 2.3;     /* Rホイール幅 */
 98:    cs->fwidth  = 0.3;     /* Fタイヤ幅 */
 99:    cs->fradius = 0.3;     /* Fタイヤ半径 */
100:    cs->rwidth  = 0.3;     /* Rタイヤ幅 */
101:    cs->rradius = 0.3;     /* Rタイヤ半径 */
102:    cs->wheelbase = 2.5;   /* ホイールベース */
103:    cs->steeringratio = 0.4 DEG;
104:                 /* ステアリング係数(マウスX座標と舵角の関係) */
105:    cs->frontdrive = 0;    /* 前輪駆動 */
106:    cs->reardrive = 1;     /* 後輪駆動 */
107:    cs->l1 = cs->wheelbase*0.5;  /* 前輪軸から重心までの距離 */
108:    cs->l2 = cs->wheelbase*0.5;  /* 後輪軸から重心までの距離 */
109:    cs->mass = MASS;       /* 質量(単位はkg) */
110:    cs->ibeta = IBETA;     /* 慣性モーメント(単位はkgf.m/s^2) */
111:
112:    zerovector( ci->b );        /* 位置(ワールド座標) */
113:    setvector( ci->ba,  1.0, 0.0, 0.0 );  /* 基底ベクトル(α軸) */
114:    setvector( ci->bb,  0.0, 1.0, 0.0 );  /* 基底ベクトル(β軸) */
115:    setvector( ci->bc,  0.0, 0.0, 1.0 );  /* 基底ベクトル(γ軸) */
116:
117:    setvector( ci->dir,  0.0, 0.0, 1.0 );
118:    setvector( ci->dirf, 0.0, 0.0, 1.0 );
119:    setvector( ci->dirr, 0.0, 0.0, 1.0 );
120:    setvector( ci->sidedir, 1.0, 0.0, 0.0 );
121:    zerovector( ci->vel );
122:    zerovector( ci->velf );
123:    zerovector( ci->velr );
124:    zerovector( ci->revf );
125:    zerovector( ci->revr );
126:    ci->revdf = 0;
127:    ci->revdr = 0;
128:    ci->scf = 1.0;
129:    ci->scr = 1.0;
130:    ci->cf = 0.0;
131:    ci->cr = 0.0;
132:    zerovector( ci->cff );
133:    zerovector( ci->cfr );
134:    ci->drvf = 0.0;
135:    ci->drvr = 0.0;
136:    ci->brkf = 0.0;
137:    ci->brkr = 0.0;
138:    zerovector( ci->trkf );
139:    zerovector( ci->trkr );
140:    zerovector( ci->tire );
141:    zerovector( ci->tiref );
142:    zerovector( ci->tirer );
143:    ci->force = 0.0;
144:    ci->ve = 0.0;
145:    ci->vd = 0.0;
146:    ci->vs = 0.0;
147:    ci->omega = 0.0;
148:    ci->rpm = 0;
149:    ci->theta = 0.0;
150:    ci->phi_fr = 0.0;
151:    ci->phi_fl = 0.0;
152:    ci->phi_rr = 0.0;
153:    ci->phi_rl = 0.0;
154:    ci->stroke_fr = cs->fradius;
155:    ci->stroke_fl = cs->fradius;
156:    ci->stroke_rr = cs->fradius;
157:    ci->stroke_rl = cs->fradius;
158:    ci->height_fr = 0.0;
159:    ci->height_fl = 0.0;
160:    ci->height_rr = 0.0;
161:    ci->height_rl = 0.0;
162:    ci->roll = 0.0;
163:    ci->pitch = 0.0;
164:    ci->heightratio = HEIGHTRATIO;
165:    return;
166: }
167:
168: void  drive( CarSpec *cs, CarInfo *ci )
169: {
170:    CarInfo  *ci0, *ci1;
171:    double  wd1, wd2;
172:    VECTOR3  wv1;
173:    VECTOR3  v1, v2, v3;
174:    int  i, ry, rd;
175:    double  co, si;
176:    int  ontime2;
177:
178:    ci0 = &ci[phase];
179:    ci1 = &ci[1-phase];
180:
181:    /* 前回の時刻との差 */
182:    ontime2 = ONTIME();
183:    dt = TIMEDIFFERENCE(ontime2,ontime)/100.0;
184:    ontime = ontime2;
185:
186:    /* ステアリング角 */
187:    ci1->theta = (double)(128-(control.mousex))*(cs->steeringrat
io);
188:
189:    /* タイヤ回転方向(実際にタイヤが向いている方向) */
190:    si = sin( ci1->theta ); co = cos( ci1->theta );
191:    scalevector( wv1, -si, ci0->ba );
192:    scalevector( ci1->revf, co, ci0->bc );
193:    addvector( ci1->revf, ci1->revf, wv1 );
194:
195:    copyvector( ci1->revr, ci0->bc );
196:
197:    /* 前輪の速度 */
198:    wd1 = -(cs->l1)*(ci0->omega);
199:    scalevector( ci1->velf, wd1, ci0->ba );
200:    addvector( ci1->velf, ci1->velf, ci0->vel );
201:    if ( normalizevector( ci1->dirf, ci1->velf ) < EPS ) {
202:      /*copyvector( ci1->dirf, si1->revf );*/
203:      ci1->revdf = 0;
204:    } else {
205:      ci1->revdf = 1;
206:    }
207:
208:    /* 後輪の速度 */
209:    wd1 = (cs->l2)*(ci0->omega);
210:    scalevector( ci1->velr, wd1, ci0->ba );
211:    addvector( ci1->velr, ci1->velr, ci0->vel );
212:    if ( normalizevector( ci1->dirr, ci1->velr ) < EPS ) {
213:      /*copyvector( ci1->dirr, ci1->revr );*/
214:      ci1->revdr = 0;
215:    } else {
216:      ci1->revdr = 1;
217:    }
218:
219:    /* スリップアングル余弦 */
220:    ci1->scf = dotproduct( ci1->revf, ci1->dirf );
221:    ci1->scr = dotproduct( ci1->revr, ci1->dirr );
222:
223:    /* タイヤの転動方向と進行方向が一致した場合
224:       内積がわずかに1を上回ることがある
225:       (コーナリングフォースの計算式のacos()がNaNを返す) */
226:    if ( (ci1->scf) >  1.0 ) ci1->scf =  1.0;
227:    if ( (ci1->scf) < -1.0 ) ci1->scf = -1.0;
228:    if ( (ci1->scr) >  1.0 ) ci1->scr =  1.0;
229:    if ( (ci1->scr) < -1.0 ) ci1->scr = -1.0;
230:
231:    /* 後退中の場合 */
232:    if ( (ci1->scf) < 0.0 ) {
233:      ci1->scf = -(ci1->scf);
234:      scalevector( ci1->revf, -1.0, ci1->revf );
235:      ci1->revdf = -1;
236:    }
237:    if ( (ci1->scr) < 0.0 ) {
238:      ci1->scr = -(ci1->scr);
239:      scalevector( ci1->revr, -1.0, ci1->revr );
240:      ci1->revdr = -1;
241:    }
242:
243:    /* タイヤ回転方向のタイヤ進行方向に垂直な成分 */
244:    scalevector( wv1, ci1->scf, ci1->dirf );
245:    subvector( ci1->cff, ci1->revf, wv1 );
246:    normalizevector( ci1->cff, ci1->cff );
247:    scalevector( wv1, ci1->scr, ci1->dirr );
248:    subvector( ci1->cfr, ci1->revr, wv1 );
249:    normalizevector( ci1->cfr, ci1->cfr );
250:
251:    /* コーナリングフォース */
252:    ci1->cf = CORNERINGFORCEF( ci1->scf );
253:    scalevector( ci1->cff, ci1->cf, ci1->cff );
254:    ci1->cr = CORNERINGFORCER( ci1->scr );
255:    /*if ( control.mouserb ) ci1->cr *= 0.8;*/
256:    scalevector( ci1->cfr, ci1->cr, ci1->cfr );
257:
258:    /* にせピッチ */
259:    ci1->pitch = 0.0;
260:
261:    /* 駆動力 */
262:    ci1->drvf = ci1->drvr = 0.0;
263:    if ( control.mouserb ) {
264:      if ( cs->frontdrive ) {   /* にせ駆動力 */
265:        ci1->drvf += (DRIVEFORCE*(MAXSPEED-(ci0->ve))/MAXSPEED);
266:        /* にせピッチ */
267:        ci1->pitch += (-PITCH*(MAXSPEED-(ci0->ve))/MAXSPEED);
268:      }
269:      if ( cs->reardrive ) {
270:        ci1->drvr += (DRIVEFORCE*(MAXSPEED-(ci0->ve))/MAXSPEED);
271:        /* にせピッチ */
272:        ci1->pitch += (-PITCH*(MAXSPEED-(ci0->ve))/MAXSPEED);
273:      }
274:    }
275:    if ( control.bskey ) {
276:      if ( cs->frontdrive ) {   /* にせ駆動力 */
277:        ci1->drvf += (-DRIVEFORCE*(MAXSPEEDR-(ci0->ve))/MAXSPEED
R);
278:        /* にせピッチ */
279:        ci1->pitch += (PITCH*(MAXSPEEDR-(ci0->ve))/MAXSPEEDR);
280:      }
281:      if ( cs->reardrive ) {
282:        ci1->drvr += (-DRIVEFORCE*(MAXSPEEDR-(ci0->ve))/MAXSPEED
R);
283:        /* にせピッチ */
284:        ci1->pitch += (PITCH*(MAXSPEEDR-(ci0->ve))/MAXSPEEDR);
285:      }
286:    }
287:
288:    /* 制動力 */
289:    ci1->brkf = ci1->brkr = 0.0;
290:    if ( control.mouselb ) {
291:      if ( (ci1->revdf) == 1 ) {
292:        ci1->pitch += PITCH;     /* にせピッチ */
293:        ci1->brkf += -BRAKEFORCEF;
294:      } else if ( (ci1->revdf) == -1 ) {
295:        ci1->pitch += -PITCH;    /* にせピッチ */
296:        ci1->brkf += BRAKEFORCEF;
297:      } else {  /* ci1->revdf == 0 */
298:        ci1->pitch += 0.0;     /* にせピッチ */
299:        ci1->brkf += 0.0;
300:      }
301:      if ( (ci1->revdr) == 1 ) {
302:        ci1->pitch += PITCH;     /* にせピッチ */
303:        ci1->brkr += -BRAKEFORCER;
304:      } else if ( (ci1->revdr) == -1 ) {
305:        ci1->pitch += -PITCH;    /* にせピッチ */
306:        ci1->brkr += BRAKEFORCER;
307:      } else {  /* ci1->revdr == 0 */
308:        ci1->pitch += 0.0;     /* にせピッチ */
309:        ci1->brkr += 0.0;
310:      }
311:      if ( (ci0->ve) < EPS ) {  /* 停止中(ややアドホック) */
312:        ci1->revdf = 0;
313:        ci1->revdr = 0;
314:        ci1->pitch = 0.0;
315:        ci1->brkr = 0.0;
316:        ci1->brkf = 0.0;
```

▶マウスパッドを裏返しにして使っていると，ゴムの面なのでかえってマウスを傷めた。
岡島　久夫(24)徳島県

```
 317:         cil->omega = 0.0;
 318:
 319:         ci0->revdf = 0;
 320:         ci0->revdr = 0;
 321:         ci0->pitch = 0.0;
 322:         ci0->brkr = 0.0;
 323:         ci0->brkf = 0.0;
 324:         ci0->omega = 0.0;
 325:       }
 326:     }
 327:
 328:     /* 前輪のトラクション */
 329:     wd1 = (cil->drvf) + (cil->brkf);
 330:     if ( (cil->revdf) == -1 ) wd1 = -wd1;
 331:     scalevector( cil->trkf, wd1, cil->revf );
 332:     /* 前輪の合力 */
 333:     addvector( cil->tiref, cil->trkf, cil->cff );
 334:
 335:     /* 後輪のトラクション */
 336:     wd1 = (cil->drvr) + (cil->brkr);
 337:     if ( (cil->revdr) == -1 ) wd1 = -wd1;
 338:     scalevector( cil->trkr, wd1, cil->revr );
 339:     /* 後輪の合力 */
 340:     addvector( cil->tirer, cil->trkr, cil->cfr );
 341:
 342:     /* 前後輪の合力 */
 343:     addvector( cil->tire, cil->tiref, cil->tirer );
 344:
 345:     /* 進行方向に向かって右向きのベクトル */
 346:     crossproduct( cil->sidedir, ci0->bb, ci0->dir );
 347:
 348:     /* 車体横方向の加速度 */
 349:     /* 前後輪の合力の大きさ+遠心力の大きさ */
 350:     cil->force = dotproduct( cil->tire, cil->sidedir ) + (cs->ma
ss)*(ci0->omega)*(ci0->ve);
 351:     cil->vs = dt*(cil->force)/(cs->mass);
 352:
 353:     /* β軸周りの力のモーメント→角速度の増分 */
 354:     cil->omega = (ci0->omega) + dt*( - dotproduct( cil->trkf, ci
0->ba )*(cs->l1)
 355:                  - dotproduct( cil->cff , ci0->ba )*(cs->l1)
 356:                  + dotproduct( cil->cfr , ci0->ba )*(cs->l2) )/(cs->
ibeta);
 357:
 358:     /* 車体進行方向の加速度 */
 359:     wd1 = dotproduct( cil->tire, ci0->dir );  /* 力の進行方向成分 */
 360:     wd2 = wd1/(cs->mass);          /* 加速度 */
 361:     /* 加速度に前回の時刻との差をかければ速度の増分が出る */
 362:     wd1 = dt*wd2;
 363:
 364:     if ( control.mouselb && ((ci0->ve) > 0.0) && ((-wd1) >= (ci0
->ve)) ) {
 365:       /* ブレーキにより停止する瞬間(ややアドホック) */
 366:       cil->vd = 0.0;
 367:       cil->vs = 0.0;
 368:       cil->omega = 0.0;
 369:
 370:       ci0->vd = 0.0;
 371:       ci0->vs = 0.0;
 372:       ci0->omega = 0.0;
 373:     } else {
 374:       /* 通常の加減速 */
 375:       cil->vd = (ci0->ve) + dt*wd2;
 376:     }
 377:
 378:     /* 車体の速度 */
 379:     scalevector( cil->vel, cil->vd, ci0->dir );
 380:     scalevector( wv1, cil->vs, cil->sidedir );
 381:     addvector( cil->vel, cil->vel, wv1 );
 382:     cil->ve = normalizevector( cil->dir, cil->vel );
 383:     if ( (cil->ve) < EPS ) {
 384:       cil->ve = 0.0;
 385:       copyvector( cil->dir, ci0->bc );
 386:     }
 387:
 388:     cil->rpm = (int)((cil->ve)*25.0 _KMPH);
 389:     if ( (cil->rpm) > 7500 ) cil->rpm = 7500;
 390:     if ( (cil->revdf) == -1 ) cil->rpm = -(cil->rpm);
 391:
 392:     cil->roll = ROLL*(cil->force)/(100.0 KGF);  /* にせロール */
 393:     if ( cil->roll < -ROLL ) cil->roll = -ROLL;
 394:     if ( cil->roll > ROLL ) cil->roll = ROLL;
 395:
 396:     if ( (cil->ve) < EPS ) {
 397:       copyvector( cil->b , ci0->b  );
 398:       copyvector( cil->ba, ci0->ba );
 399:       copyvector( cil->bb, ci0->bb );
 400:       copyvector( cil->bc, ci0->bc );
 401:     } else {  /* 進行 */
 402:       /* 車体を回転する */
 403:       co = cos( (cil->omega)*dt );
 404:       si = sin( (cil->omega)*dt );
 405:       /* 位置ベクトル */
 406:       scalevector( wv1, dt, cil->vel );
 407:       addvector( cil->b, ci0->b, wv1 );
 408:       /* 基底ベクトルはそのものを回転する */
 409:       /* bb(β軸)は旋回運動では変化しない */
 410:       scalevector( wv1,     co, ci0->ba );
 411:       scalevector( cil->ba, si, ci0->bc );
 412:       addvector( cil->ba, cil->ba, wv1 );
 413:
 414:       scalevector( wv1,    -si, ci0->ba );
 415:       scalevector( cil->bc, co, ci0->bc );
 416:       addvector( cil->bc, cil->bc, wv1 );
 417:
 418:       copyvector( cil->bb, ci0->bb );
 419:
 420:       /* 衝突チェック */
 421:       checkcollision( cs, ci );
 422:
 423:       /* 検索起点 */
 424:       scalevector( wv1, -(cs->height)*0.5, cil->bb );
 425:       addvector( wv1, wv1, cil->b );
 426:       i = checks2( &ry, &rd, collide_checkinfolist, (int)(wv1[0]
 _SLASHUNIT), (int)(wv1[1] _SLASHUNIT), (int)(wv1[2] _SLASHUNIT) );
 427:       if ( i != -1 ) {
 428:         /* 壁にぶつかったら前の状態に戻す */
 429:         if ( collide_polygonlist->polygon[i].palet == TP_BACKGRO
UND ) {
 430:           copyvector( cil->b , ci0->b  );
 431:           copyvector( cil->ba, ci0->ba );
 432:           copyvector( cil->bb, ci0->bb );
 433:           copyvector( cil->bc, ci0->bc );
 434:         }
 435:         /* 新しい位置(y座標のみ変わる) */
 436:         cil->b[1] = (double)ry SLASHUNIT;
 437:         /* 面法線を求めて新しいβ軸とする */
 438:         wv1[0] = -(double)collide_checkinfolist->ci[i].a;
 439:         wv1[1] = -(double)collide_checkinfolist->ci[i].b;
 440:         wv1[2] = -(double)collide_checkinfolist->ci[i].c;
 441:         normalizevector( cil->bb, wv1 );
 442:         /* 新しいγ軸(補正前のγ軸から新β軸成分を取り除いて正規化したもの) */
 443:         wd1 = dotproduct( cil->bb, cil->bc );
 444:         scalevector( wv1, wd1, cil->bb );
 445:         subvector( wv1, cil->bc, wv1 );
 446:         normalizevector( cil->bc, wv1 );
 447:         /* 新しいα軸(新β軸と新γ軸の外積) */
 448:         crossproduct( cil->ba, cil->bb, cil->bc );
 449:       }
 450:
 451:       /* タイヤの転動 */
 452:       wd1 = (cil->vd)*dt*((double)(cil->revdf));
 453:       /* 前左 */
 454:       wd2 = (-si*(cs->fshaft)/2 + co*(cs->l1) - (cs->l1)) + wd1;
 455:       cil->phi_fl = ci0->phi_fl + wd2/(cs->fradius);
 456:       /* 前右 */
 457:       wd2 = ( si*cs->fshaft/2 + co*(cs->l1) - (cs->l1)) + wd1;
 458:       cil->phi_fr = ci0->phi_fr + wd2/(cs->fradius);
 459:
 460:       wd1 = (cil->vd)*dt*((double)(cil->revdr));
 461:       /* 後左 */
 462:       wd2 = (-si*(cs->rshaft)/2 - co*(cs->l2) + (cs->l2)) + wd1;
 463:       cil->phi_rl = ci0->phi_rl + wd2/(cs->rradius);
 464:       /* 後右 */
 465:       wd2 = ( si*cs->rshaft/2 - co*(cs->l2) + (cs->l2)) + wd1;
 466:       cil->phi_rr = ci0->phi_rr + wd2/(cs->rradius);
 467:
 468:       /* タイヤの高さ */
 469:       /* 前左 */
 470:       scalevector( v1, -(cs->fshaft)/2, cil->ba );
 471:       scalevector( v2,  -(cs->fradius), cil->bb );
 472:       scalevector( v3,           cs->l1, cil->bc );
 473:       addvector( wv1, cil->b, v1 );
 474:       addvector( wv1, wv1, v2 );
 475:       addvector( wv1, wv1, v3 );
 476:       i = checks2( &ry, &rd, collide_checkinfolist, (int)(wv1[0]
 _SLASHUNIT), (int)(wv1[1] _SLASHUNIT), (int)(wv1[2] _SLASHUNIT) );
 477:       cil->height_fl = ry SLASHUNIT;
 478:       if ( i == -1 ) {
 479:         cil->stroke_fl = (cs->fradius);
 480:       } else {
 481:         cil->stroke_fl = (rd SLASHUNIT)*(cil->bb[1]);
 482:       }
 483:       /* 前右 */
 484:       scalevector( v1,  (cs->fshaft)/2, cil->ba );
 485:       scalevector( v2,  -(cs->fradius), cil->bb );
 486:       scalevector( v3,           cs->l1, cil->bc );
 487:       addvector( wv1, cil->b, v1 );
 488:       addvector( wv1, wv1, v2 );
 489:       addvector( wv1, wv1, v3 );
 490:       i = checks2( &ry, &rd, collide_checkinfolist, (int)(wv1[0]
 _SLASHUNIT), (int)(wv1[1] _SLASHUNIT), (int)(wv1[2] _SLASHUNIT) );
 491:       cil->height_fr = ry SLASHUNIT;
 492:       if ( i == -1 ) {
 493:         cil->stroke_fr = (cs->fradius);
 494:       } else {
 495:         cil->stroke_fr = (rd SLASHUNIT)*(cil->bb[1]);
 496:       }
 497:       /* 後左 */
 498:       scalevector( v1, -(cs->fshaft)/2, cil->ba );
 499:       scalevector( v2,  -(cs->fradius), cil->bb );
 500:       scalevector( v3,          -cs->l2, cil->bc );
 501:       addvector( wv1, cil->b, v1 );
 502:       addvector( wv1, wv1, v2 );
 503:       addvector( wv1, wv1, v3 );
 504:       i = checks2( &ry, &rd, collide_checkinfolist, (int)(wv1[0]
 _SLASHUNIT), (int)(wv1[1] _SLASHUNIT), (int)(wv1[2] _SLASHUNIT) );
 505:       cil->height_rl = ry SLASHUNIT;
 506:       if ( i == -1 ) {
 507:         cil->stroke_rl = (cs->fradius);
 508:       } else {
 509:         cil->stroke_rl = (rd SLASHUNIT)*(cil->bb[1]);
 510:       }
 511:       /* 後右 */
 512:       scalevector( v1,  (cs->fshaft)/2, cil->ba );
 513:       scalevector( v2,  -(cs->fradius), cil->bb );
 514:       scalevector( v3,          -cs->l2, cil->bc );
 515:       addvector( wv1, cil->b, v1 );
 516:       addvector( wv1, wv1, v2 );
 517:       addvector( wv1, wv1, v3 );
 518:       i = checks2( &ry, &rd, collide_checkinfolist, (int)(wv1[0]
 _SLASHUNIT), (int)(wv1[1] _SLASHUNIT), (int)(wv1[2] _SLASHUNIT) );
 519:       cil->height_rr = ry SLASHUNIT;
 520:       if ( i == -1 ) {
 521:         cil->stroke_rr = (cs->fradius);
 522:       } else {
 523:         cil->stroke_rr = (rd SLASHUNIT)*(cil->bb[1]);
 524:       }
 525:     }
 526:     phase = 1 - phase;
 527:
 528:     return;
 529: }
```

▶いま一番ほしいもの……それは単位。いま一番いらないもの……それは試験。単位がほしいよ～。
伊南　尚幸(19)山形県

SIDE B
自動車工学入門

Yokouchi Takeshi 横内 威至

**いよいよ自動車工学の分野へ足を踏み入れる
そこで，車の姿勢制御に活躍するサスペンションに焦点を当て
その仕組みから車の挙動を探っていく**

最近「リッジレーサー2」が面白い。路面が滑りやすくなっている，というより，滑っていてもなぜか姿勢変化のための力がかけられる驚異のグリップがたまらない。コーナーをクリアするときに1回転スピン（？）していくのがトレンドだ。だけど，なんか車を運転してる感じじゃないよな。現実にはあんな運転ができるわけないからね。そんなわけで，ということでもないが，自動車工学，というより力学の理解を深めていきたい。まあ，独学なのでもしかしたら理解ではなく誤解が含まれている危険性を秘めているが。

いままでに丹氏が行ってきたように，ドライビングシミュレータを作ろうとすると，いろいろと面倒なことが多い。また，最終的にアセンブラで記述するつもりなのでリストの掲載は非常に厳しい。いろいろとプログラムを溜め込んであるのだが，それだけでもかなりの量になってしまっているのでどうしようもないだろう。とりあえず，理論だけでもしっかりと確立できるようにしていこうと思う。

ドライビングシミュレータというからには，車の挙動をできる限り再現させなくてはならない。そのためには数式で車の挙動を表し，あらゆる設定を数値によって行う必要がある。まず洗い出す必要があるのは，挙動自体。次に，挙動に関わるパラメータのうちまとめられるものはまとめ，いらないものは省いていく。どんなものをパラメータとして扱うべきか，これをしっかりと考えなければならない。

たとえば，エンジンであるが，緻密に考えるとシリンダーだとかマニフォールドだとか，あらゆる構成部品の影響を考えなければならない。しかし，実際の走行に必要なパラメータは回転数やトルクといった出力に関するデータである。当然，セッティングによってこれらのグラフは変化するが，それはセッティングを変化させたときに再計算してグラフを作ればいい。エンジンセッティングのパラメータが100種類あるとしても，走行中に必要なパラメータはほんの数種類ですむ。

まずは，この数種類のパラメータ，走行に関わる最低限かつ最大限に簡略化したパラメータを見つけ出すことが重要な目的となる。このあたりのバランスを極力重視したい。サスペンション，ブレーキでの変化，荷重移動によるグリップの変化，それによるテールスライドやドリフトなんかはセッティングしだいでどうにでも変化させられるのが望ましい。

4輪接地モデルのための準備

車の挙動をとらえるとき，基本的には重心の運動方程式と姿勢を考えることになる。ただし，これは特異な状況での話である。というのは平面上，常に道路が平面であり，しっかりと4輪が地面についた状態での話なのである。当然，そのような状況が望ましいのだが，むしろそうでない状況のほうが多い。坂道であれば前後のタイヤにかかる荷重移動を計算しなければならないし，加速，減速でもそうだ。当然，コーナリングならば遠心力によって姿勢が崩れ，しっかりと荷重移動を計算しなければならない。

まず車を走らせるためには，なにを考えればよいのだろうか。基本的に車は，アクセルを踏み込めば加速し，ブレーキを踏めば減速する。どういうことかというと，アクセルでエンジンの燃料噴射量，回転数を高めてタイヤに駆動力を作らせることである。ブレーキはタイヤの回転に力を加えて止めようとして制動力を作らせる。この駆動力，そして制動力を地面から車体に伝えるのがタイヤの役割だ。

では，悪質なデコボコの道路ではタイヤはどうなっているだろう。もし車体にタイヤが固定されていれば，場合によってはタイヤが地面に接していない状況も起こり得る。当然，そのようなタイヤからは車の挙動に変化を与える力は一切消し去られてしまう。車としては4輪からしっかりと力を受けることが望ましいし，下手にどれかのタイヤが路面から

離れてしまえば，車は危険な挙動を起こすことになりかねない。

そんな危険な状況を回避するため，車にはタイヤからこれらの力を可能な限り有効に伝えられるように，車体とタイヤの間にバネのようなシステムが入っている。これがサスペンションである。4輪と車体の姿勢を考えるには，このサスペンションの役割をしっかりと把握しなければならない。

サスペンション

いろいろなシミュレータが，サスペンションの効果を取り入れていないことに気づいているだろうか。コーナーではロールしないし，当然タイヤが浮き上がるようなこともない。サスペンションが強烈に硬い，といってもやっぱり無理がある。確かにサスペンションのような複雑なシステムは考えたくない。車体とタイヤの間にスプリングがあり，そして4本とも独立しているので，もし真面目に考えれば非常に複雑な方程式を組まなければならないからだ。

簡単なモデルとして，バネの上下にオモリをつけたものを考えよう。上のオモリを固定し，下のオモリを引っ張って離すとバネは上下に減衰振動を繰り返す。そして，スパッと上のオモリを離し，下のオモリだけを固定すると，今度は上のオモリを基準として減衰振動を起こす。さらに，上下のオモリを一切固定しないときにはどうなるだろうか。高校以上の方ならば覚えているかもしれない。物理が嫌になる分野のひとつであり，途中計算は複雑を極める部分である。サスペンションはまさにこれの応用で，上のオモリに相当するのが車体。バネはそのオモリの4カ所につけられ，タイヤや路面の影響が下のオモリに相当する。こんな異常なモデルは極悪を究める大学入試でも見掛けないであろう。

では省くことにするか？　いや，まずは考えてみなければならない。サスペンションは車の挙動において重要な働きをもっているのだから。

サスペンションの効果は，実際に車に乗っていればすぐにわかる。「乗り心地」なんてのはサスペンションの効果そのものであろう。「軟らかい」サスペンションならば，段差を乗り越えるときに衝撃が小さく，フワッと乗り越えていく。逆に「硬い」サスペンションならばガツッと大きい衝撃とともに，瞬間的に越える。高級なリムジンだとかは快適な乗り心地が要求されるため，当然衝撃が弱くなるように作られるだろう。スポーツカーならば常に路上の状況が把握できるよう，そして長い間車体が不安定に揺れる状況を嫌うためにも硬めのサスペンションが採用されているはずだ。このように「硬い」「軟らかい」のような表現がなされるが，実際にはどのようなものであろうか。また，それによって乗り心地以外にどのような変化が起こるだろうか。大体，乗り心地とはどのような影響で感じられるものなのだろうか。

サスペンションについて詳しく調べれば，それだけで1冊の本になるほどである。莫大な量になってしまうのでそこまで詳しくは扱わない。しかし，細かい挙動こそセッティングの妙味。可能な限りの影響を取り入れるためにひととおり考えてみよう。

まず，サスペンションの役割をはっきりさせておくと，

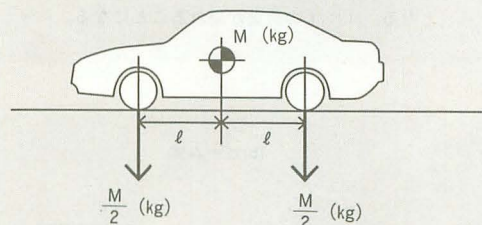
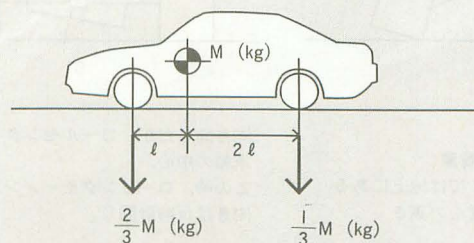
1）車体姿勢を作る
2）タイヤを接地させ，トラクションを作る
3）外力の影響など，衝撃を吸収する

などが考えられる。このうち，シミュレーションで特に重要なのは1）と2）。3）に関しては取り入れることが難しい。バネによって衝撃を吸収するのだが，現在可能なシミュレーションでは1ターンにかかる周期が特定できない。減衰振動のシミュレーションということになるのだが，滑らかな変化を見せなければかえって気分の悪いものになるであろう。

地面に固定したモデル

まず，簡単なモデルの車をいろいろな場所に置くことを考えよう。

たとえば図1のような状態である。水平な道路に

図1　重心と荷重

前輪と後輪に等しい荷重がかかる。
さらに左右輪について，重心が中心ならば$\frac{M}{4}$（kg）ずつの荷重となる。

前輪，後輪に2：1の配分で荷重がかかる。
前後ともに同じバネなら前輪の沈み方は後輪の2倍になる。

車を置けば，理想的な重量配分となり，姿勢については理解しやすい状態となる。サスペンションがすべて同じならば重心の位置によって自然な姿勢になるであろう。

次は図2であるが，サスペンションで前後がつながったようなモデルは切り捨ててしまってかまわない。おそらくそんな車は存在していないと思う。前後の傾斜によって荷重が前，あるいは後ろに移動するだけであり，前輪，後輪への荷重が変化すると考えればいい。このときの重量配分は，また別のところで考える。

では図3である。右カーブを走行中ならばこのような荷重となるであろう。慣性によってコーナーの外側に荷重が移動するような状況であり，コーナー外側，図3では左側のタイヤに重量が多目にかかるようになる。実際の車をみるとさらに複雑な状況だ。まず，サスペンションの方式によって車体の傾き方が違う。基本的にロールセンターを中心に車体が傾き，それによってサスペンションにかかる荷重は変化するのだが，2つのサスペンション形式を比べてみよう。

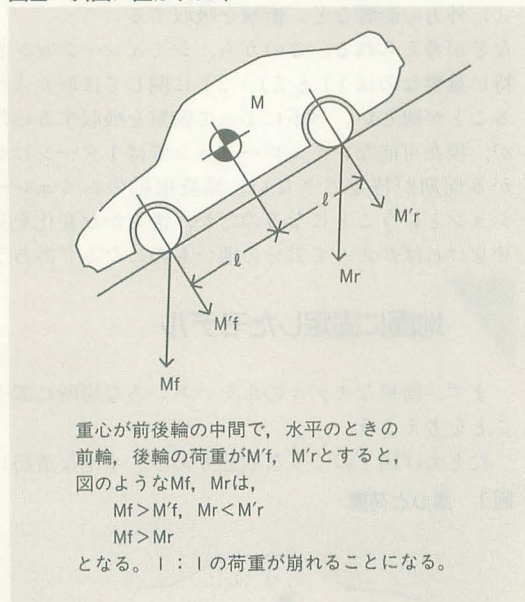



図3は(a)が独立懸架，(b)がビーム式のサスペンションでの運動である。どちらもコーナーの外側に傾くが，ロールセンターが(a)は地上，(b)はシャフトの中心にあるとする。当然図のようにローリングモーメントは違ったものとなる。車体重量が同じならば(a)のほうが大きいモーメントがかかることになり，ロールは(a)が大きくなる。コーナー内側のスプリングは伸びるので，車体を元に戻そうとするし，同時に限界ならばタイヤを持ち上げようとする。外側のタイヤは車体を押し返そうとすると同時に，タイヤを路面に押しつけることになる。これが結果的にどうなるかというと，内側のグリップが弱まり，外側のグリップが強まることになる。

また，独立懸架のときで，重心が高いとどうなるであろう。(c)のように，タイヤの接点より外側に荷重をかけるように力が働き，いずれは転覆することになる。これだけでは独立懸架が不利に見えるが，独立懸架は大幅な軽量化が可能であり，またいろいろな要因があるのでこれだけで有利不利は決定できない。

そして，サスペンションの形態はこんな単純な分類ではない。もっと複雑な機構のものである。ここ

図2　斜面に置かれた車

重心が前後輪の中間で，水平のときの
前輪，後輪の荷重がM′f，M′rとすると，
図のようなMf，Mrは，
　Mf＞M′f，Mr＜M′r
　Mf＞Mr
となる。1：1の荷重が崩れることになる。

図3　サスペンション

(a)独立懸架

a：向心加速度（m/s²）
M：車重（kg），厳密には質量
O：ロールセンター，ここでは地上にある
h：ロールした状態での重心の高さ

このとき，車は遠心力F＝－Maを受ける。
よってローリングモーメントはMah。
向きは反時計回りである。

(b)ビーム式

(a)と同じだが，ロールセンターは
車軸の中心。
この時，ローリングモーメントはMah′。
向きは反時計回り。

同じコーナリングでもモーメントは違う。
h＞h′なので，この場合(a)のほうがロールが
大きくなる。

(c)
重力による力　F_1
遠心力　F_2

この2つの力の合力はA点に作用する。
モーメントはA点に対して反時計回りに
働くので，A点を支点として転覆する。

で独立懸架のロールセンターを地上としたが，これはあくまで一例である。図4のようにロールセンターは考えられる。しかし，このロールセンターは瞬間回転中心でしかない。状況によって図4（b）のようにロールセンターは移動する。一般の自動車工学によれば，ロール角が大きくならないうちはロールセンターは固定と考えて差し支えないらしいので，とりあえず固定という考えに従うことにしよう。

これで基本的なサスペンションの動作が理解できたと思う。では，ここまでのポイントを洗い出す。まず重心の位置，大きさ，そしてロールセンターである。そして姿勢を決定するにはさらに左右のスプリングのパラメータが必要である。これに遠心力や重力を考慮して姿勢を決定しなければならない。サスペンションの形態で変化するのはロールセンターと，左右のスプリングの変化の割合となる。もうひとつ，左右タイヤの間隔，トレッドというパラメータがここから先は必要となる。ここまでで，荷重移動を含めた数式による解釈を図5に示してみよう。

固定モデルの致命点

以上，車がある瞬間にどのような力を受け，そしてどのような姿勢をとっているかを考えてみた。しかし，これをシミュレーションに応用しようとすると非常に困ったことになる。

たとえば，車が時速5kmで段差に乗り上げたときどのような挙動を起こすだろうか。図6（a）のように，サスペンションはほとんど伸縮を起こさず，段差に乗り上げるだろう。では時速100kmではどうか。（b)のように乗り上げた瞬間にサスペンションが大きく縮み，そのあと車体を持ち上げるような動作となる。明らかに車の挙動は別のものになる。

シミュレーションでは1ターンごとに，微小ではあるが一定時間ごとの姿勢を求めることになるので，一定時間ごとの姿勢しか計算されない。途中経過がどうであろうとまったく関係ないのである。途中が段差だろうと坂であろうと，次の瞬間の姿勢は同じなのである。つまり，計算はされても画面に現れることがないのだ。

シミュレーションが難しいのは，結局はこのせいなのである。連続した動作を時分割して先を計算する方法を見つけ出さねばならないのである。そうでないにしても，この例では車全体の重量を分割，つまり最低限バネ上重量，バネ下前輪重量，バネ下後輪重量に分割し，これらをバネで接続，それぞれの運動方程式までも考えなければならない。この処理は4輪接地モデルを考えるときにうまくごまかす必要がある。どうやって処理するかは，また来月ということにする。

図4　ロールセンター

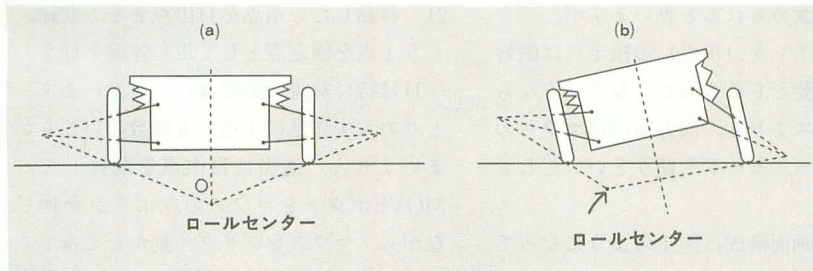
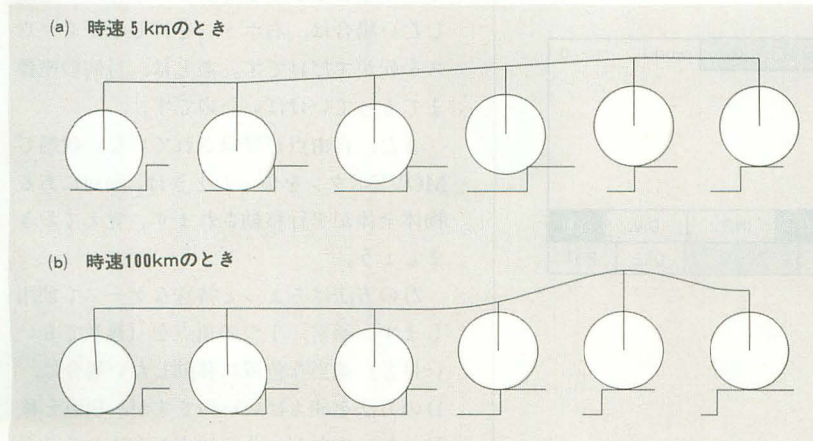
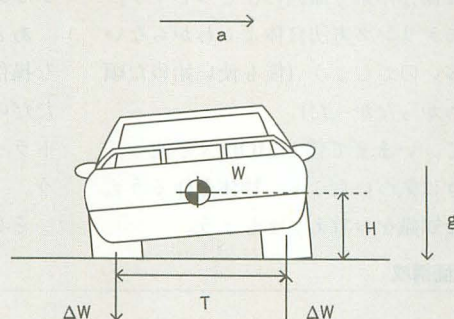



図5　荷重移動の計算

a：向心加速度（m/s^2）
H：重心高（m）
W：車重（質量）(kg)
T：トレッド（m）
ΔW：荷重移動量（kg）
g：重力加速度（m/s^2）
遠心力　F＝Wa（N）
ロールモーメントM＝FH＝WaH（N・m）

$$\Delta W \cdot g\ (N) = \frac{WaH}{T}\left(\frac{N \cdot m}{m}\right)$$

ただし，前輪を無視。Wは後輪にかかる車重とする。
たったこれだけで荷重移動の計算ができる。

図6　サスペンションの動き

MOD.Xのやさしいマニュアル

モデリングを楽しんでみよう

Hamazaki Masaya 浜崎 正哉

複雑怪奇な操作体系，馴染みの薄いモデリング方法
便利そうだけど，なんかとっつきにくそうなMOD.X
そこで，楽しくモデリングするためのテクニックを紹介しましょう

用意されたプリミティブ（基本形状）を加工して，思いどおりの3Dポリゴンモデルを作ることを目指したSLASH用モデラMOD.X。10月号の付録ディスク「もみじ狩りPRO-68K」に新バージョンが発表されました。前回配布したときと違い，今回は実行ファイルも収録されていましたので，多くの人が実際に動かしてみたことと思います。僕がデザインした形状データのサンプルも収録されているので，MOD.Xを使えばどのようなものが作れるか，だいたいわかったことでしょう。

しかし，機能が増えてずいぶんと便利にはなりましたが，3面図に慣れ親しんでいる人や初めて触る人にとっては，なかなか思いどおりにいかないかもしれません。複雑怪奇な操作体系も原因のひとつですが，やはりモデリング方法自体よくわからない部分が多いのでしょう（僕も使い始めた頃は全然わからなかった）。

そこで，いままで僕がMOD.Xを使ってきたときに気づいたこと，物体を作るうえで必要な知識をお教えしましょう。

基本姿勢

MOD.Xでは，キーボードとマウスを併用して作業を進めていくことになります。2つのものを同時に使うことは，結構面倒臭い面もありますが，ショートカットキーを使いこなせば，作業効率がずいぶんと上がるはずなのでよしとしましょう。併用なんて面倒な操作はいやだ～，とだだをこねてもキーボードのみに割り当てられている機能もあるので，いまのところは「しようがない」とあきらめてください。

まあ，このキーボードのみ，あるいはマウスのみでMOD.Xを使うことができないことは，多少問題があるので，今後のバージョンでは改められると思いますが。

あと，ポインタ（頂点）の指定には微妙な操作が必要とされるので，なるべくならただのマウスよりも，前方後円墳マウスのトラックボールモードを使うといいでしょう。

そして，画面構成は図1のようになっています。それぞれ，マウスでどのような操作を行う必要があるかも書いておきましたので，参考にしてください。

図1　画面構成

伸ばして縮めて動かそう

それでは，MOD.Xで物体を作るためのもっとも基本的なテクニック，伸ばして縮めて移動してみます。

まず，頂点を移動させるためには，2つの方法があります。

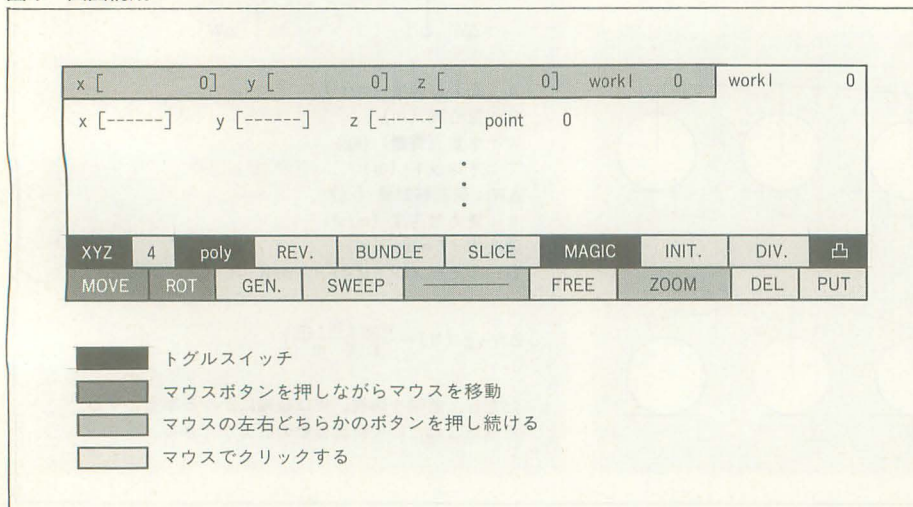
1)　頂点を選択し，MOVEボタンで移動する

・左ボタン＋左右方向＝X座標の移動
・左ボタン＋上下方向＝Z座標の移動
・右ボタン＋上下方向＝Y座標の移動

2)　移動したい頂点を自由点として登録。もう1点を固定点として辺の伸縮を行う

1)は特に解説の必要はないと思います。とりあえず任意の1点でも複数の点でもかまいません。適当に自由点を登録して，MOVEボタンをマウスの左ボタンを押しながら，マウスをグリグリ動かしてみてください。X，Z方向に頂点がごろごろ移動するのを確認できるでしょう。Y方向に移動したい場合は，右ボタンを押しながらマウスを転がすだけです。あとは，目的の座標までもっていけばいいのです。

また，自由点が登録されていない状態でMOVEボタンを使ったときは，画面にある物体全体が平行移動されます。覚えておきましょう。

2)の方法はちょっと特殊なケースで使用します。通常，1つの頂点を（複数でもいいけど）適当な位置に移動したい場合は，1)の方法を使えばいいのですが，頂点を移動したい方向が，その頂点が含まれる辺の延長上にある場合です。だいたい，垂直方

向ならいざしらず，任意の辺のベクトルの正確な移動先などわかりはしません（だいたいはわかるでしょうが）。そこで，基本となる辺の１点を固定点とし，辺の伸縮を行うことで，辺のベクトル方向へ頂点を移動させることができます（図２）。

また，頂点の座標を知りたい場合は，指点（黄色）を定義しておきましょう。指点の設定方法は，頂点にマウスカーソルを合わせるだけです。すると画面の上から２行目に，その指点の座標が表示されるようになります。

次に辺を伸縮，移動させてみましょう。といっても基本的に頂点の移動と原理は同じです。適当な２点（ある辺に属していなくても辺として登録できる）をつなぎ，移動，辺の伸縮，移動を行うだけ。ポリゴンに含まれる辺を登録したときには白線，ポリゴン以外の辺を登録したときは肌色の線で結ばれることも覚えておきましょう。

あと，連続していない辺を複数登録したい場合には，１辺を登録したあとに別の頂点で右クリック（コネクト情報の消去）を行ってください。でないとよけいな辺まで結んでしまいます。

そうそう，頂点，辺の移動が確定したらいったんFREEボタンをクリックして，すべての頂点を解放しておくクセをつけておきましょう。よけいな操作ミスを招くことが少なくなりますよ。

座標固定機能も使おう

頂点を移動させるためにマウスを転がしていると，ふとした拍子によけいな方向へ動くときがあります。そんな誤動作を防止するために用意されたのが座標固定機能です。これは，画面左下にある「XYZ」の文字をクリックすることで反転し，その反転した座標の移動をキャンセルさせることができます。特にＸ方向のみ，Ｚ方向のみに移動させたいときに積極的に活用してください。

図2　頂点Aを$\overline{AB}$方向へ移動したい場合

また，全部を選択してしまうと，物体は現在ある座標に固定されてどうにも動かなくなります（なぜか辺の伸縮だけはできたりするけど）。当然のことながら無意味ですので，やらないほうがいいでしょう。

あと，座標固定機能のテクニックとして，簡単に直方体が作れます。

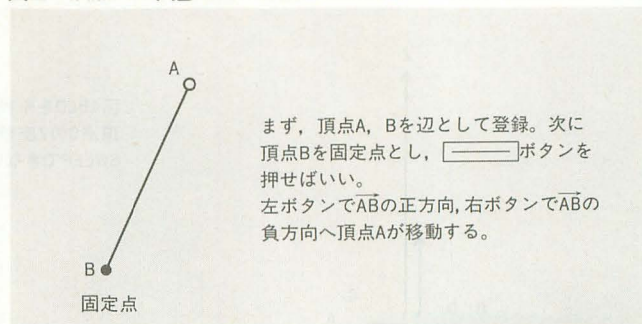
まず，移動させたい座標以外を固定します。そして，ZOOMボタンを押すだけ。固定した以外の方向に伸びたり縮んだりします。しかも，座標固定機能を使えば２軸方向へ同時に変形させることもできるので，非常にお得です。ぜひ覚えてください。

ちなみに，座標固定機能を使わない場合は，移動したい座標に属する辺をすべて選択してから，辺を伸縮させなければなりません。どちらの方法が簡単かすぐにわかるでしょう。

回してみる

回転はROTボタンで行えるのですが，これについては特に説明しません。回転操作は頂点移動と同じで，選択された頂点の重心を回転中心として回転が行われることを理解しておけば問題ありません。

多面体を作る

それでは，MOD.Xの動作がだいたい理解できたところで，具体的に物体を作成することを考えます。立方体，四面体，三角柱を作成するときは，Ｏキーを押してメニュー一発で生成できるので問題ありません（写真１）。

しかし，メニューにないような形状を作成したい場合は，どのようにすればいいのでしょうか。いきなり飛行機だとか車だとかを作成する前に，簡単な例として五角柱を作ってみます。

まず最初に立方体を生成します（写真２）。そして，ある１面を選択し，SWEEPボタンを押します。すると，選択した面と側面の間に新しく面が生成されますので，辺の伸縮を使って新しい側面を引き出します(写真３)。ある程度引き出したら，２辺を選択して同時に縮めるのです。で，２辺の頂点どうしがくっつけば，見事五角柱の出来上がりです(写真４)。あとは好きなように頂点の座標を調節して，思いどおりの

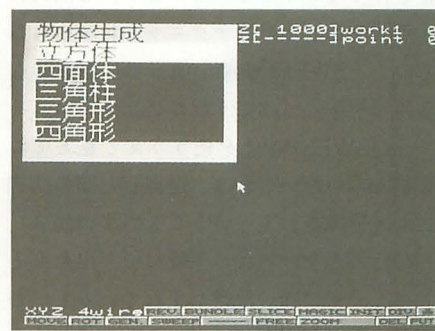



写真１

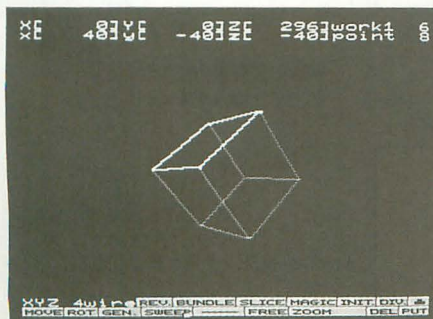



写真２

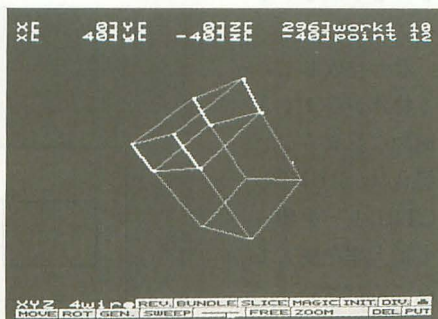



写真３

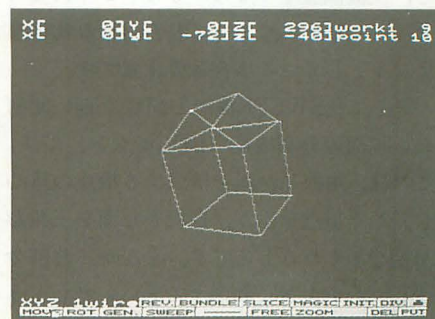



写真４

図3 SWEEP機能

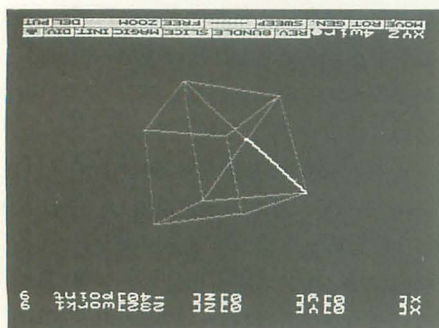
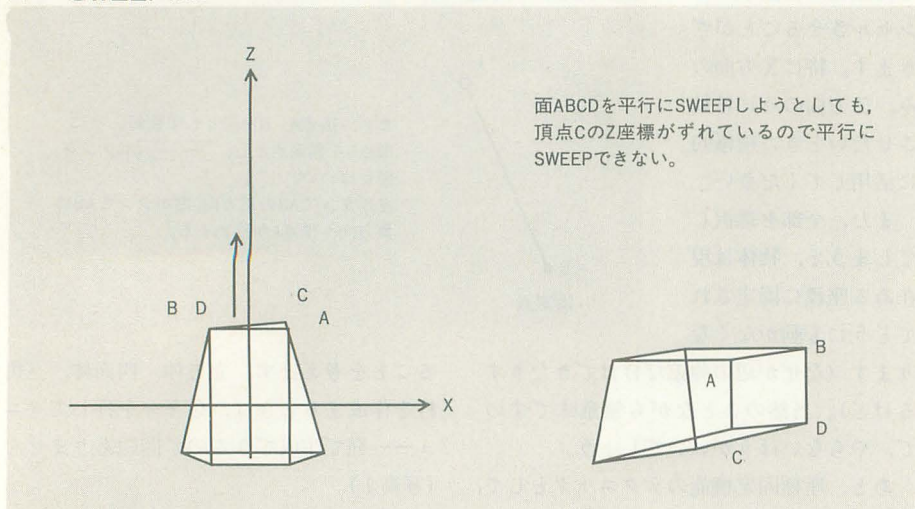



写真5

形を作ってください。なお，六角柱なら2辺を縮めるだけで完成します。

結局，MOD.Xを使ってモデリングするときには，多面体をうまく組み合わせて物体を作っていくことになります。必然的に多面体を作り上げるテクニックが求められるので，SWEEP機能を使っていろいろな多角形を作ってみてください。

またSWEEP機能を使ううえで覚えておいてもらいたいことは，選択した面の法線方向へ伸縮されるということです（図3）。このため面を座標軸に平行に引き出そうとしても，元の面の座標がずれていると，わずかに方向がずれます。こうなってしまうと，位置調整をするのがものすごく大変になってしまいます。特に左右対称な物体を作ろうとしている場合は要注意です。

そして，SWEEP機能と同じくらい便利なのが面の分割機能（DIV.ボタン）です。これは，選択した面を構成する頂点の重心に新しく頂点を作り，選択した面を三角形分割してくれるものです。とんがった部分を作りたいときに利用するといいでしょう（写真5）。

3面図の活用

MOD.Xは，基本的に透視図を使って物体を作成していきます。ユーザーは透視図にプリミティブを置き，加工することで任意の形状を作成しなくてはなりません。ただし，用意されているプリミティブは大きさが決まっていますし，置かれる位置も原点に固定されています。しようがないのでユーザーは，目的の位置までプリミティブを移動させてから加工しなくてはなりません。はっきりいって，この操作は非常に面倒なものです。

そこで，任意の位置に任意のポリゴンを生成できる3面図モードが用意されています(図4)。基本的に指定できる座標は3面図のグリッド上(方眼)のみ（右クリックで座標指定),形状は四角形までが原則です。4点以上設定しても動きますが，正常動作はしません。MOD.Xをあまりいじめないようにね。

なお作成している物体がうまく3面図に収まらない場合は，SHIFTとCTRLキーを使って拡大率を変えてみてください。

さらに，この3面図では回転体を生成するための任意の伸縮辺を設定できます。伸縮辺は，

・X軸を回転軸とする場合は，XZ平面
・Y軸を回転軸とする場合は，XY平面
・Z軸を回転軸とする場合は，YZ平面

に設定してください。

ちなみに「多面体を作る」のところで，面を引き出して……という方法で五角柱を作成していました。しかし，SWEEP機能を使わなくても，正五角柱を作りたいなら回転体生成機能を使えば一発で生成可能です。ただし，回転体生成機能で分割数を細かく設定しすぎると，あっという間に面数が増えますので注意しましょう。ついでにいうと，あまり変な伸縮辺を設定すると破綻をきたして暴走します。

合成してみる

MOD.Xでは物体編集ワークを3つもっています（F1,F2,F3キーで切り替え)。これは3個，別々のオブジェクトを同時に編集することができ，それぞれの編集ワーク

図4 3面図モード

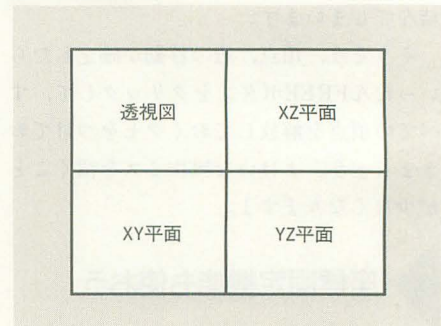
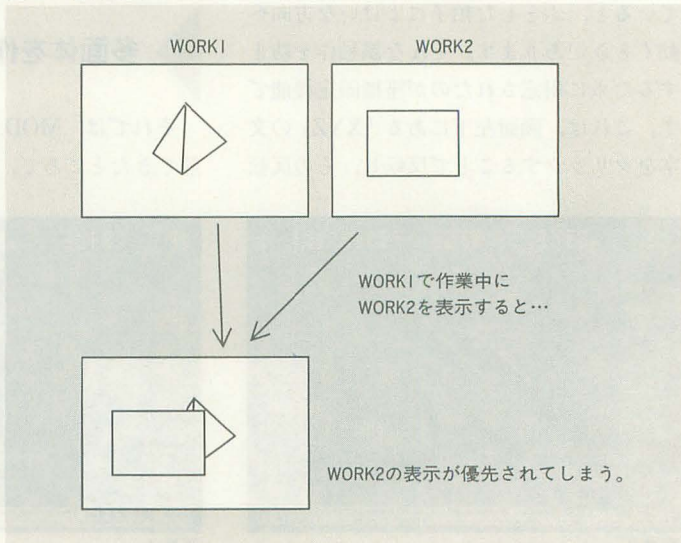



図5 物体編集ワーク

ごとに合成もできます（Mキー）。つまり，1つのオブジェクトを作るためにパーツ(部品)ごと，別々の物体編集ワークで作業を進めることができるのです。うまく編集ワークを切り替えて，作業効率を上げてください。

あと，それぞれの編集ワークの物体も表示できるようになっています（F6,F7,F8キー）。位置合わせの確認に使うといいでしょう。しかし，困ったことに別編集ワークで編集している物体に，現在編集している物体が隠れてしまう場合が多々あります(図5)。これは，現在編集している編集ワークの物体が一番先に描画されてしまうのが原因です（文句をつけるのを忘れていた）。

しようがないので，いまのところはいったん別の物体を編集している編集ワークに切り替えて，位置確認を行ってください。もしくはワイヤーフレーム表示モードで対処するしかないでしょう。

思いどおりの物体を作る

それでは，いよいよ本格的に物体をモデリングしてみましょう。ここでは，オリジナルの車をデザインしてみます(写真6)。

実際にMOD.Xで作業を始める前に，頭の中で作りたいと思っている物体のラフを描きます(図6)。いきなりフロント部分のみ，かつとんでもなくいい加減なラフ描きですが，作業の進め方を決めることができれば用が足ります。自信がなければきちんと3面図を描いておくのもいいでしょう。

作業の手順を説明していきます。まず立方体を置きます(写真7)。ボンネット部分を作るため，ちょっと縦方向を潰してから，どんどんSWEEPしていきます(写真8)。曲面っぽくするために3段階に分けてSWEEPしているのを確認してください（写真9）。

次に横方向へSWEEPします（写真10)。ここでは，頂点がきちんと重なっているかどうかの確認と，重なり合っている部分に発生する余分なポリゴンを削除することを忘れないように(図7)。あとは同じ要領で3段階ほど横方向にSWEEPして，もっこりとした感じを出します(写真11)。これでフロント部分はOK。

今度はコクピット部分を作るため後方に

写真6

図6 ラフ描き

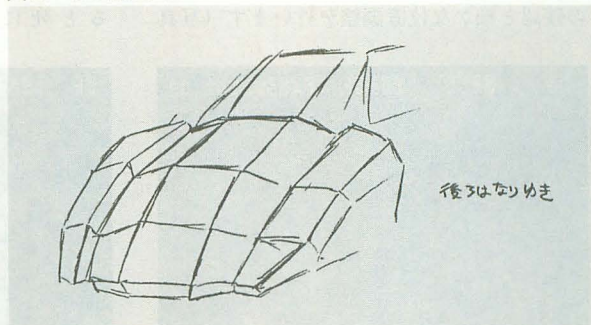



写真7

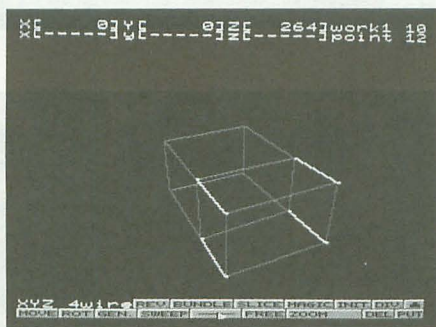



写真8

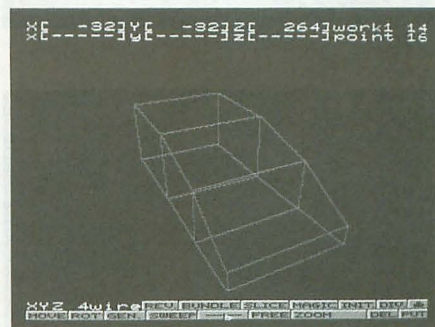



写真9

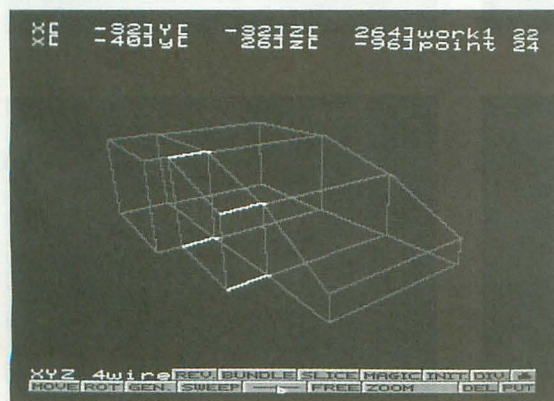
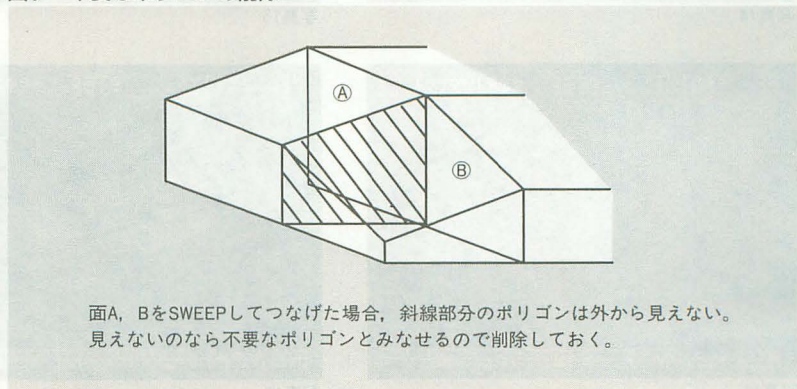



写真10

図7 不要なポリゴンの削除

面A，BをSWEEPしてつなげた場合，斜線部分のポリゴンは外から見えない。見えないのなら不要なポリゴンとみなせるので削除しておく。

SWEEPします(写真12)。横方向にドアもどきの出っ張りを作ってから，さらにリア部分のために後方へSWEEPしておきます(写真13)。そして，中央部分を上にSWEEPしてから少し縮め，さらに色を変えてフロントガラスを作ります(写真14)。ドアの部分のガラスは，ちょいちょいと頂点を結び，GEN.ボタンでポリゴンを生成するだけ(写真15)。ここでも隠れてしまう無駄なポリゴンを消去しておくのを忘れないように。

あとは，バックガラス(?)を作るために辺を後方に移動して(写真16)，色を変えます(写真17)。お次はリアタイヤのカバーといえる部分を作るために，後方のブロックを横にSWEEPします(写真18)。ちょっと丸みのあるデザインにしたいので，2段階くらいSWEEPします(写真19)。

以上で，基本形状が完成。3面図で全体の確認と細かな位置調整を行います(写真20)。完全に対称形に作れたな，と思ったところで，おもむろに対称生成機能を使います(写真21)。見事に車ができました。

あとは，全体のバランスを考えて，さらに位置調整をし，ランプの部分の色をちょっと変えます(写真22)。最後に，底面の無駄なポリゴンをガシガシ消去しておきましょう(写真23)。

以上の作業をもって車が完成です(タイヤがないけど)。ひとつの立方体が，みるみるうちに変化して立派な車になりました。モデリング作業自体はだいたい4時間ぐらい。あちこち細かい修正で結構時間が食われましたが，あれだけいい加減なラフから始めたにしては，そこそこの時間で完成したといえるでしょう(満足)。

あと，いくらSLASHが速いからといって，これほどなにも考えずにモデリングすると死にます。実際，X68000 XVI(16MHz)を使って作業を行ったのですが，最後のほうでは結構しんどいものがありました。皆さんは，面数が増えすぎないように気をつけてください。

あと対称生成機能についてひと言。対称生成機能を使う場合は，対称軸にまたがっているポリゴンに注意してください。頂点座標が完全に対称となっていないと，ほとんど似たような位置へポリゴンが定義されてしまいますからね。

そのほかの機能

以上で説明してきた機能以外にも，MOD.Xにはたくさんの機能がありますが，特に解説はしません。僕自身すべての機能を活用しているわけではありませんので，まだ，残りの機能をどのように活用したらいいか把握できていないのです。

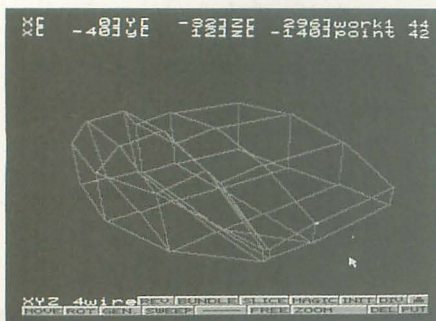



写真11

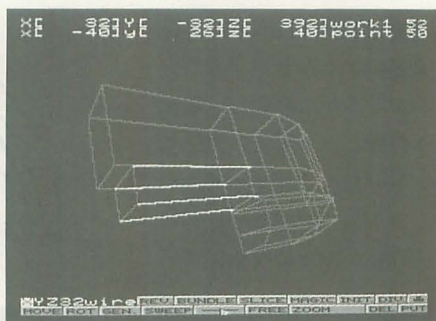



写真12

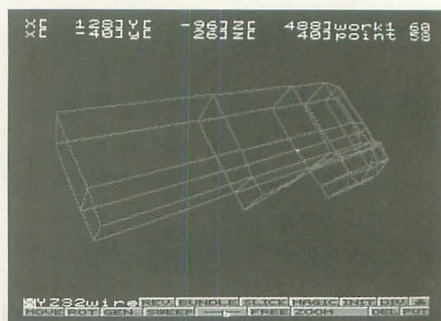



写真13

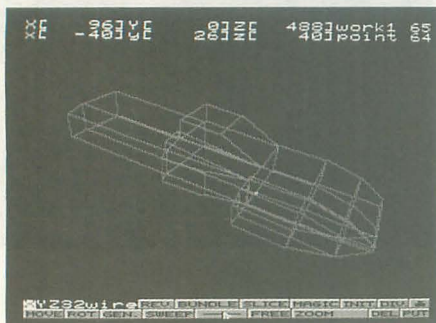



写真14

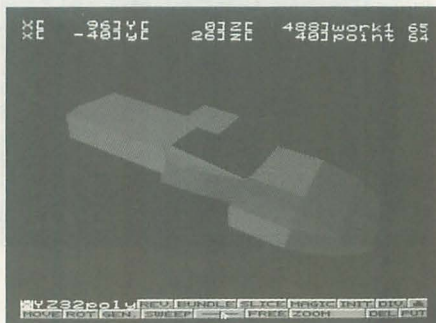



写真15

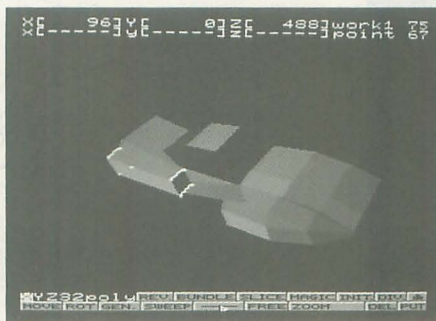
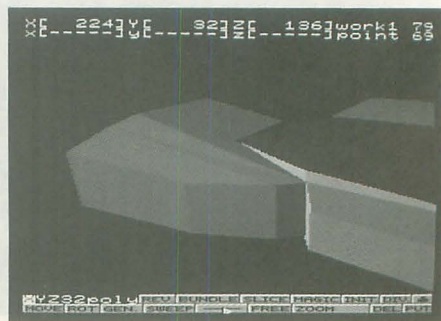
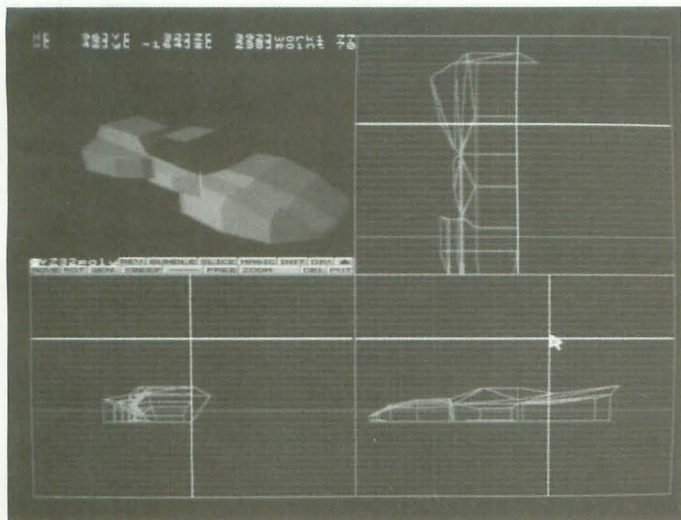
写真16

写真17

写真18

写真19

写真20

写真21

しかし，これではちょっと情けないので，ここではポリゴンをスライスするときの注意をいくつか書いておきます。

まず，魔法板を出現させたあとは，MOVE,ROTボタンは魔法板のみに作用しますので，物体を編集したいときには，いったん魔法板を消去するようにしてください。

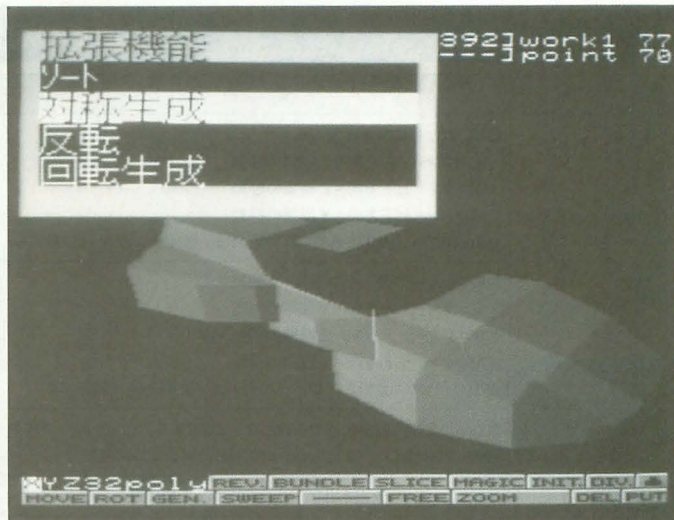
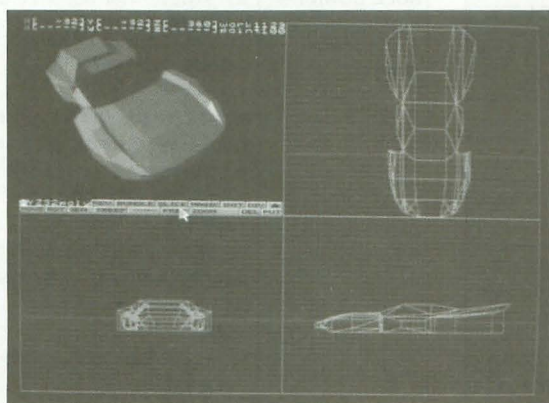
写真22

写真23

次に物体をスライスしたときは，多少の誤差が生じます(整数演算のため)。いらないポリゴンを削除したい場合は，BUNDLEで物体全体の辺を登録し，必要なポリゴン部分のみ登録辺を解除してから，DELボタンを押してマメに消去したほうがいいでしょう。

一応，本文最後にMOD.Xの機能一覧表をまとめておきましたので，どのような機能があるか確認してみてください。なお，一覧表では，ショートカットキーは明記されていません。

慣れれば快適?

ここまで説明してきたことを理解できれば，あなたもMOD.Xでばっちりモデリングができるはずです。しかし，慣れないうちは，かなり操作ミスをするはずですから，ある程度まで作業が進んだらいったんセーブしておくクセをつけましょう。

MOD.Xを使うときに難しいことは，なにかひとつのことをするのでも複数の解決方法がある点です。これについては，実際に触って自分なりに解釈していくしかありません。今回の記事では，僕なりの使い方ということでいろいろ書きましたが，僕の行っている方法が最良というわけではありません。それなりに経験は積んでいるはずなので，結構役に立つだろうとは思ってますけど。

ちょっとだけ触ってみて投げ出してしまった人も，とりあえずこの記事を読んで概要をつかむことができたら，もう一度MOD.Xを立ち上げてみてください。きっとMOD.Xがやさしく思えることでしょう。

あと，まだ表面化していないようですが，MOD.Xにはバグがあちらこちらにあります。ファイル関係は結構怪しいし，固定点の扱いもちょっとおかしくなるときがあります(いきなり固定点が原点に飛ぶ)。さらにX68030では，キャッシュ(命令，データともに)をOFFにしないと誤動作します。セーブはマメにしておきましょう。

皆さんもなにかバグを見つけたら，Oh!X編集部までご連絡ください。なるべく詳しい状況を書いてくれると嬉しいですね。もちろん，MOD.Xへの要望も受けつけていますから，なにかほしい機能があれば遠慮なくアンケートハガキでお聞かせください。

そうそう，なにやらMOD.Xで作成したデータを，DōGA CGAシステムでも使いたいという要望がありましたね。しかし，CGAシステムの形状データは公開されているし，「.PLG」ファイルもダンプすれば，データ形式など一発でわかるでしょう。簡単だと思うんですけどね。また，逆にCGAシステムの形状データをMOD.Xにもってこられるとおいしいかもしれません。要するに相互データコンバータってやつですか。あると便利かもしれませんね。うむ，気が向いたら作ってみますか。

それでは，MOD.X，そしてSLASHを思う存分活用してください。

ひと目でわかるMOD.X機能一覧表

ボタン（キー）	機能
MOVE	登録された自由点を移動する。 ・左ボタン+左右方向＝X座標の移動 ・左ボタン+上下方向＝Z座標の移動 ・右ボタン+上下方向＝Y座標の移動 自由点がなにも登録されていない場合は，物体全体が平行移動する。
ROT	登録された点の重心を中心にして回転する。 ・左ボタン+左右方向＝X座標の回転 ・左ボタン+上下方向＝Z座標の回転 ・右ボタン+上下方向＝Y座標の回転 自由点がなにも登録されていない場合は，物体全体が回転する。
XYZ	反転した文字の座標の移動を行わないようにする。X，Z方向への移動制御，ZOOMボタンでの拡大縮小の方向制御のために使う。
SWEEP	登録したポリゴンを引き出す。多面体を作るために使う。
――	辺の伸縮をする。辺は複数登録してもいい。左ボタンで登録した辺のベクトルの正方向（ <―> ），右ボタンで負方向へ伸縮される（ >―< ）。SWEEP機能でポリゴンを引き出すときにも使う。
FREE	現在，登録されている自由点，固定点，指点のすべてを開放する。座標移動を行ったあとに，実行しておくと無駄な操作ミスを防げる。
ZOOM	物体全体の拡大，縮小を行う。座標固定機能も併用できる。
BUNDLE	登録された辺を含む物体のすべての辺を登録する。
DIV.	登録されたポリゴンを三角形分割する。ポリゴンは三角形でも四角形でも可。
GEN.	3点もしくは4点で結ばれた頂点にポリゴンを生成する。登録されている辺は閉じている必要あり。3面図を使えば任意の位置にポリゴンを生成できる。また，4点登録した場合，自動的に三角形分割を行う。
REV.	登録されているポリゴンの法線方向を逆にする。GEN.ボタンでポリゴンを生成したとき，反対方向に向いているポリゴンを表示させるときなどに使う。
DEL	登録されているポリゴンの削除を行う。登録されているポリゴンが複数ある場合は，繰り返して実行することにより順次削除されていく。
凸／凹	頂点の移動により四角形のポリゴンが破綻したときの回避方法を指定する（トグルスイッチ）。凸のときは出っ張った感じ，凹のときはへこんだ感じで三角形分割を行う。
MAGIC	ポリゴンを切断するための魔法板のON/OFF制御を行う（トグルスイッチ）。
SLICE	魔法板で分割された断面でポリゴンを切断する。切断された片方のポリゴン群は自由点で構成されている辺で囲まれる。ただし，移動前にFREEボタンで頂点を開放してしまうと，分離されずに頂点が固定されてしまうことに注意。
INIT.	魔法板を初期化する。
4	移動，伸縮，回転のための基本移動量の切り替えを行う（初期値は4）。左クリックで増えていき，右クリックで減っていく。設定範囲は1～64まで。
wire（poly）	編集している物体のワイヤーフレーム，ポリゴン表示を切り替える（トグルスイッチ）。XF5キーでも同様の操作ができる。
Oキー	プリミティブを生成する。生成できるプリミティブは，立方体，四面体，四角柱，三角形，四角形の5種類。プリミティブは原点に生成される。
Fキー	物体のロードを行う。ファイル名を指定したい場合は，UNDOキーを押してからファイル名を入力する。なお，入力ファイルは「.PLG」の拡張子のファイルのみ（入力の必要なし）。
Sキー	物体のセーブを行う。ファイル名を指定したい場合は，UNDOキーを押してからファイル名を入力する。なお，出力ファイルは「.PLG」の拡張子が自動的につく（入力の必要なし）。

ボタン（キー）	機能
F1，F2，F3キー	物体編集ワークの切り替え。 ・F1キー……物体編集ワーク1 ・F2キー……物体編集ワーク2 ・F3キー……物体編集ワーク3
F6，F7，F8キー	現在編集している物体編集ワーク以外の表示をON/OFFする（トグルスイッチ）。ただし，他物体編集ワークのエディットはできない。 ・F6キー……物体編集ワーク1のON/OFF ・F7キー……物体編集ワーク2のON/OFF ・F8キー……物体編集ワーク3のON/OFF
Mキー	現在編集している物体をほかの物体編集ワークの物体と合成する。
Xキー	辺で囲まれたポリゴンを別の物体編集ワークへコピーする。コピーするポリゴンは複数でも可。
ROLL UPキー	登録されたポリゴンの優先順位を下げる。ポリゴン番号が小さいほど優先順位が低くなる。
ROLL DOWNキー	登録されたポリゴンの優先順位を上げる。ポリゴン番号が大きいほど優先順位が高くなる。
Eキー	拡張機能メニューの呼び出し。 ・ソート 物体に定義されたポリゴンをソートする。基本的に作られた順番に優先順位が上がっていくので，後ろから物体を作っていくと前からはちゃんと見える物体ができる。どうしようもないときは，ROLL UP，ROLL DOWNキーで直接優先順位を変えること。 ・対称生成 XY，YZ，XZ平面に対称なポリゴンを生成する。対称軸にまたがるポリゴンの頂点座標は，きっちり揃えておかないとよけいなポリゴンが定義されてしまうので注意。 ・反転 XY，YZ，XZ平面に対して，座標を反転させる。 ・回転生成 3面図に設定された伸縮辺に従って，回転体を生成する。ただし，あまり変な伸縮辺を設定すると暴走することあり。また，あまり複雑な伸縮辺，分割数を設定するとポリゴン数が増えすぎるので注意。
Aキー	XZ平面のグリッド表示のON/OFFを行う（トグルスイッチ）。
TABキー	透視図，3面図モードの切り替え（トグルスイッチ）。
CTRL，SHIFT	3面図モードでの物体表示の拡大縮小を行う。CTRLキーで拡大，SHIFTキーで縮小。
Cキー	登録されたポリゴンの色情報を変更する。複数ポリゴンを登録することも可。なお，プリミティブを生成するときは，生成前にカラー設定を行わないと，以前に設定されたカラーコードで物体が生成されてしまうことに注意（起動時はstd_black）。
CLRキー	現在作業している物体編集バッファの初期化を行う。
HOMEキー	視点を初期状態に戻す。
カーソル，OPT.1，OPT.2キー	現在編集している物体の表示方向を変える。 ・←→キー……HEAD方向（Y軸） ・↑↓キー……PITCH方向（X軸） ・OPT.1，OPT.2キー……BANK方向（Z軸）
2，4，6，8，XF1，XF2キー	現在編集している物体の表示座標を変える。 ・4，6キー……X座標の移動 ・2，8キー……Y座標の移動 ・XF1，XF2キー……Z座標の移動
Pキー	辺で囲まれたポリゴンと同形状で大きさが1/4のポリゴンを張りつける。
Qキー	モデラを終了する。確認メッセージは出るが，物体編集中でも容赦なく終了してしまうので注意。

注）I（辺で囲まれたポリゴンを分離する）は，動作不安定のため省略してあります。バグフィクスが行われるまで使用しないでください。

もうCAD.Xなんていらない？

MC_CNV.BAS

Hamazaki Masaya 浜崎 正哉

ずいぶんと進化したMOD.X。せっかく筋のいいポリゴンモデラなんだから，SLASHのみで使うのはもったいない。そこで，MOD.Xの形状データをCGAシステムの形状データに変換するコンバータを作ってみました。

とりあえずバージョン0.0

気が向いたら作る，とほとんど冗談半分で書いた「MOD.X←→CGAシステム」形状データコンバータ。時間的に余裕があるわけではなかったのですが，なんとなく作りはじめてしまったら，とりあえず形になってしまいましたので発表します。

形になったといっても，まだ「.PLG」ファイルを「.SUF」ファイルに変換するだけです。しかも，対応しているのは，サーフェイスモデルのみ。しかし，これでMOD.XでモデリングしたDōGA CGAシステムにもっていけるようになりました。CAD.Xでしかできないようなこともありますが，面倒なモデリングなどはMOD.Xでラクラクできるでしょう。

「.PLG」ファイルの中身は？

まず，ターゲットとなるMOD.Xの形状データファイル「.PLG」のフォーマットを調べます。作った本人に聞けばいいのですが，あいにく連絡がとれませんでした。しようがなくMOD.Xでセーブした「.PLG」ファイルをダンプしてみます。

すると，ひと目でわかるヘッダ部分があり，頂点データらしきデータ列とポリゴン定義用の頂点コネクト情報が続いているようです。

図1 「.PLG」ファイルのデータ形式

ヘッダ（24バイト）
頂点個数（2バイト）
頂点データ（頂点個数×6バイト）
ポリゴン枚数（2バイト）
コネクト情報 ・三角形の場合 識別子（＝0，2バイト） 頂点番号（6バイト） カラー番号（2バイト） ・四角形の場合 識別子（＝1，2バイト） 頂点番号（8バイト） カラー番号（2バイト）

ここで，ふむふむとうなずきながらさらに推測してみます。

「まず，頂点個数情報が1ワード，XYZ座標それぞれ1ワードとして1頂点データは6バイトだな。で，ポリゴン枚数情報が1ワード，コネクト情報が三角形の場合は3ワード？　それとも2ワード？　四角形の場合は？」

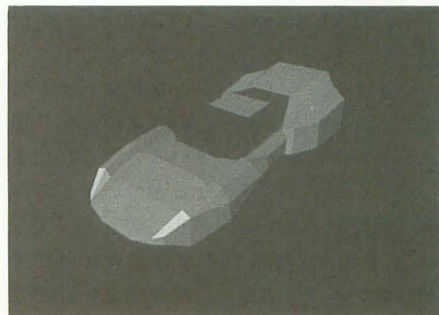
結局，三角形1枚，そして四角形1枚を定義した「.PLG」ファイルをダンプしてみたところ，図1のようなフォーマットであることがわかりました。

コンバート作業

データ構造がわかったらコンバート作業は楽なものです。まず，頂点情報を取り出し，コネクト情報に従ってポリゴンデータを出力していけばいいのです。

しかし，問題がひとつありました。それは四角形の処理です。SLASHでは，四角形の4頂点が同一平面上になくても無理やり描画させることができます。MOD.Xでは，一応こういった面の破綻を回避してくれますが，チェックが甘く，わずかにずれた座標でも四角形として登録できるのです。

しかし，CGAシステムではそうはいきません。頂点がきちんと同一平面上にないポリゴンは，認識してくれないのです。認識してくれないものはしようがない。結局，四角形は三角形2枚に分割して出力するようにしました。分割方法はいたって簡単。頂点番号1,2,3,4で囲まれたポリゴンがあったとします。すると頂点番号1,2,4と2,3,4の2つの三角形に分割できるのです。

本来ならば，きちんと4頂点が同一平面上に乗っているかのチェックが必要なのですが，時間の関係上はしょっています。次のバージョンでは，きちんと対応します。

CGAシステムでレンダリング

このコンバータの使い方はいたって簡単。まず，BASICを立ち上げてからRUNするだけ。あとは，コンバートする「.PLG」ファイルを入力すれば，自動的に「.SUF」ファイルが出来上がります。なお，ファイル名の入力時には，拡張子を省略してください。

で，出来上がった形状データはさっそくレンダリングしてみたいというのが人情というものです。ここでは，表示テスト用のフレームファイル（リスト2），タイムチャート（リスト3），標準アトリビュートデータ（リスト4）を用意しました。フレームファイルとタイムチャートにある形状名は，各自作成したものに書き換えてください。あとは，

REND TEST.SUF STD.ATR TEST.FSC

HANIM TEST

とすれば，クルクル回るアニメーションを観賞できます。

アトリビュートデータは，かなり適当に作ったものですので，気に入らなかったら各自手を加えてみてください。用意されているのはMOD.Xで使われている「std_black」から「std_white」までの標準16色のみです。なお，カラー番号16以上を指定した場合は，アトリビュートの場所に

「COL??」と設定されます。

本当は，SLCOL.IDX（MOD.Xのカラー定義ファイル）を覗けばカラーネームはわかるのですけどね。あと，ベタカラーには対応していません。

相互コンバータへ

以上，非常に低機能ですが，リストも短いのでちゃっちゃか打ち込んで活用してくれると嬉しいですね。

ところで，なぜ，時間もないのにいきなりコンバータ制作作業に手をつけたかというと，「XL/Image」でCGAのデータが使える，ということがあったためです。

いままで，レイトレーシングシステムにはあまり興味がなかったのですが，先日届いたサンプルをいじっていると，これがまた楽しいんです。そうなると，いままで作ってきたデータをレンダリングして，美しいCGを作成してみたくなるじゃないですか。リアルタイム処理の醍醐味もいいのですが，超美麗な3D画像もなかなかいいなあ，なんて思ってしまったわけです。

要するに，やりたいことには労力は惜しまない性格なんで（その代わり興味のないことは全然やらないけど），近いうちにもっときちんとしたバージョンをお届けしたいと思います。

リスト1　出力例

```
obj suf plane1 {
/* cad line color = 2 */
atr blue
prim poly  (   -40        24        17
                40        40        25
               -40       -24        17 )

prim poly  (    40        40        25
                40       -40        25
               -40       -24        17 )

/* cad line color = 14 */
atr white
prim poly  (   -40        24       -17
               -40        24        17
              -296         0         0 )

prim poly  (   -40       -24       -17
               -40        24       -17
              -296         0         0 )

prim poly  (   -40       -24        17
               -40       -24       -17
              -296         0         0 )

prim poly  (   -40        24        17 '
               -40       -24        17
              -296         0         0 )

prim poly  (   -40        24       -17
               -40       -24       -17
                40        40       -25 )

prim poly  (   -40       -24       -17
                40       -40       -25
                40        40       -25 )

/* cad line color = 4 */
atr darkred
prim poly  (    40        40       -25
                40       -40       -25
                40        40        25 )

prim poly  (    40       -40       -25
                40       -40        25
                40        40        25 )

/* cad line color = 4 */
atr red
prim poly  (   -40       -24       -17
               -40       -24        17
                96      -168         0 )

prim poly  (    40       -40       -25
               -40       -24       -17
                96      -168         0 )

/* cad line color = 12 */
atr yellow
prim poly  (    40       -40        25
                40       -40       -25
                96      -168         0 )

/* cad line color = 4 */
atr red
prim poly  (   -40       -24        17
                40       -40        25
                96      -168         0 )

prim poly  (   -40        24       -17
                96       168         0
               -40        24        17 )

prim poly  (    40        40       -25
                96       168         0
               -40        24       -17 )

/* cad line color = 12 */
atr yellow
prim poly  (    40        40        25
                96       168         0
                40        40       -25 )

/* cad line color = 4 */
atr red
prim poly  (   -40        24        17
                96       168         0
                40        40        25 )

}
```

リスト2　テスト表示用フレームファイル

```
#frame ( fno,1,20 )
@4.2@
fram {
        light pal ( rgb ( 1 1 1 ) -6 3 -4 )
        { mov ( 19530.47 0.00 9765.24 ) eye deg ( 60 ) }
        {   rotz ( ¥div(0,360,1,21,fno)¥ )
                mov ( 2712.00 0.00 1675.20 )
                obj test }
        { target }
        }
#endframe
```

リスト3　テスト表示用タイムチャート

```
.timechart
test[ 1 - 20 ]
.endchart
```

リスト4　標準アトリビュートデータ

```
/*
        Attribute Editor ver3.0
*/
atr  black      {
                        col ( rgb ( 0.15 0.15 0.15 ) )
                        tra ( 0.00 )
                        amb ( 0.60 )
                        dif ( 0.75 )
                        spc ( 0.40 0.75 0.60 )
                }
atr  darkgray   {
                        col ( rgb ( 0.30 0.30 0.30 ) )
                        tra ( 0.00 )
                        amb ( 0.60 )
                        dif ( 0.75 )
                        spc ( 0.40 0.80 0.65 )
                }
atr  darkblue   {
                        col ( rgb ( 0.15 0.15 0.50 ) )
                        tra ( 0.00 )
                        amb ( 0.45 )
                        dif ( 0.75 )
                        spc ( 0.40 0.85 0.65 )
                }
atr  blue       {
                        col ( rgb ( 0.35 0.35 1.00 ) )
                        tra ( 0.00 )
                        amb ( 0.25 )
                        dif ( 0.75 )
                        spc ( 0.40 0.75 0.60 )
                }
atr  darkred    {
                        col ( rgb ( 0.50 0.15 0.15 ) )
                        tra ( 0.00 )
                        amb ( 0.30 )
                        dif ( 0.75 )
                        spc ( 0.40 0.85 0.60 )
                }
atr  red        {
                        col ( rgb ( 1.00 0.30 0.30 ) )
                        tra ( 0.00 )
                        amb ( 0.20 )
                        dif ( 0.85 )
                        spc ( 0.35 0.75 0.60 )
                }
atr  darkmagent {
                        col ( rgb ( 0.50 0.30 0.50 ) )
                        tra ( 0.00 )
                        amb ( 0.20 )
                        dif ( 0.85 )
                        spc ( 0.35 0.75 0.60 )
                }
atr  magenta    {
                        col ( rgb ( 1.00 0.40 1.00 ) )
                        tra ( 0.00 )
                        amb ( 0.20 )
                        dif ( 0.85 )
                        spc ( 0.35 0.75 0.60 )
                }
atr  darkgreen  {
                        col ( rgb ( 0.15 0.50 0.15 ) )
                        tra ( 0.00 )
                        amb ( 0.20 )
                        dif ( 0.85 )
                        spc ( 0.35 0.75 0.60 )
                }
atr  green      {
                        col ( rgb ( 0.30 1.00 0.30 ) )
                        tra ( 0.00 )
                        amb ( 0.20 )
                        dif ( 0.85 )
                        spc ( 0.35 0.75 0.60 )
                }
```

▶つ，ついにハードディスクを買ったぞ。340Mバイトで……明るい未来が僕を待ってる。……は，マネーがない。
山崎　聖一(18)愛知県

```
   atr  darkcyan   {
                        col ( rgb ( 0.15 0.50 0.50 ) )
                        tra ( 0.00 )
                        amb ( 0.20 )
                        dif ( 0.85 )
                        spc ( 0.35 0.75 0.60 )
                   }
   atr  cyan       {
                        col ( rgb ( 0.35 1.00 1.00 ) )
                        tra ( 0.00 )
                        amb ( 0.20 )
                        dif ( 0.85 )
                        spc ( 0.35 0.75 0.60 )
                   }
   atr  darkyellow {
                        col ( rgb ( 0.50 0.50 0.15 ) )
                        tra ( 0.00 )
                        amb ( 0.20 )
                        dif ( 0.85 )
                        spc ( 0.35 0.75 0.60 )
                   }
   atr  yellow     {
                        col ( rgb ( 1.00 1.00 0.30 ) )
                        tra ( 0.00 )
                        amb ( 0.20 )
                        dif ( 0.85 )
                        spc ( 0.35 0.75 0.60 )
                   }
   atr  lightgray  {
                        col ( rgb ( 0.75 0.75 0.75 ) )
                        tra ( 0.00 )
                        amb ( 0.20 )
                        dif ( 0.85 )
                        spc ( 0.35 0.75 0.60 )
                   }
   atr  white      {
                        col ( rgb ( 1.00 1.00 1.00 ) )
                        tra ( 0.00 )
                        amb ( 0.25 )
                        dif ( 0.85 )
                        spc ( 0.35 0.75 0.60 )
                   }
```

リスト5 MC_CNV.BAS

```
  10 /*
  20 /* MOD.XのデータをCAD.Xで使えるようにするプログラム
  30 /*
  40 /*                MC_CNV.BAS
  50 /*
  60 dim int point_data(3000)  /*頂点座標格納用配列
  70 dim int cn(3)             /*コネクト情報
  80 dim int cw(3)             /*コネクト情報
  90 int fp1,fp2               /*ファイルポインタ
 100 int point_cnt             /*頂点個数
 110 int poly_cnt              /*ポリゴン枚数
 120 int p1,p2,p3
 130 int old_c=9999
 140 str fn,dat
 150 dim str atr_name(15)={"black","darkgray","darkblue",
 160             "blue","darkred","red","darkmagent",
 170             "magenta","darkgreen","green",
 180             "darkcyan","cyan","darkyellow",
 190             "yellow","lightgray","white" }
 200 /*
 210 /* メイン
 220 /*
 230   fn=""
 240   while fn=""
 250     input "変換ファイル名",fn
 260   endwhile
 270   fp1=fopen(fn+".PLG","r") /*読み込みファイルオープン
 280   fseek(fp1,24,0)          /*ヘッダ部分の読み飛ばし
 290   point_read()             /*頂点座標の読み込み
 300   fp2=fopen(fn+".SUF","c") /*書き出しファイルオープン
 310   poly_cnt=word_read()     /*ポリゴン枚数の読み込み
 320   convert()                /*コンバート作業
 330   fcloseall()
 340 end
 350 /*
 360 /* コンバート作業
 370 /*
 380 func convert()
 390 int i,w,tp
 400    fwrites("obj suf "+fn+" {",fp2)
 410    cr_put()
 420    for i=0 to poly_cnt-1
 430       w=word_read()        /*ポリゴンタイプチェック
 440       if w=0 then tp=2     /*3角形の場合
 450       if w=1 then tp=3     /*4角形の場合
 460       prim_out(tp)
 470    next
 480    fwrites("}",fp2)
 490    cr_put()
 500 endfunc
 510 /*
 520 /* プリミティブ出力
 530 /*
 540 /*     fputc(&h09,fp2)
 550 func prim_out(type)
 560 int i,c
 570 str num
 580    for i=0 to type
 590      cn(i)=word_read()   /*コネクト情報の取り出し
 600    next
 610    c=word_read()          /*色情報の読み込み
 620    if old_c <> c then {
 630      if c >= 2 then {
 640         fwrites("/* cad line color = "+itoa(c/2*2)+" */",fp2)
 650         cr_put()
 660      }
 670      if c<=15 then{
 680         fwrites("atr "+atr_name(c),fp2)
 690      } else {
 700         fwrites("atr "+"COL"+itoa(c),fp2)
 710      }
 720      cr_put()
 730    }
 740    old_c=c
 750    if type=2 then {
 760      cw(0)=cn(0)
 770      cw(1)=cn(1)
 780      cw(2)=cn(2)
 790      tri_out(type)
 800    } else {
 810      div_tri()
 820    }
 830 endfunc
 840 /*
 850 /* N角形を出力
 860 /*
 870 func tri_out(type)
 880 int i
 890      fwrites("prim poly  (",fp2)
 900      for i=0 to type
 910        data_make( point_data(cw(i)*3+2))
 920        fwrites(dat,fp2)
 930        fwrites(" ",fp2)
 940        data_make(-point_data(cw(i)*3  ))
 950        fwrites(dat,fp2)
 960        fwrites(" ",fp2)
 970        data_make(-point_data(cw(i)*3+1))
 980        fwrites(dat,fp2)
 990        if i<>type then {
1000          cr_put()
1010          fwrites("                 ",fp2)
1020        }
1030      next
1040      fwrites(" )",fp2)
1050      cr_put()
1060      cr_put()
1070 endfunc
1080 /*
1090 /* 文字列作成
1100 /*
1110 func data_make(d)
1120 str w
1130 int sp,i
1140     dat=""
1150     w=itoa(d)
1160     sp=6-len(w)
1170     if sp<>0 then {
1180       for i=0 to sp-1
1190         dat=dat+" "
1200       next
1210     }
1220     dat=dat+w
1230 endfunc
1240 /*
1250 /* 4角形を3角形分割する
1260 /*
1270 func div_tri()
1280       cw(0)=cn(0)
1290       cw(1)=cn(1)
1300       cw(2)=cn(3)
1310       tri_out(2)
1320 /*
1330       cw(0)=cn(1)
1340       cw(1)=cn(2)
1350       cw(2)=cn(3)
1360       tri_out(2)
1370 endfunc
1380 /*
1390 /* 頂点データ読み込み
1400 /*
1410 func point_read()
1420 int i
1430   point_cnt=word_read()     /*頂点個数の読み出し
1440   for i=0 to point_cnt*3-1
1450       point_data(i)=word_read()
1460   next
1470 endfunc
1480 /*
1490 /* ファイルからワードデータの読み出しを行う
1500 /*  (ちくしょう!なんでX-BASICには
1510 /*                 ワードサイズの変数型がないんだ!)
1520 /*
1530 func word_read()
1540 int dt=0
1550    dt=fgetc(fp1)
1560    dt=dt*256
1570    dt=dt+fgetc(fp1)
1580    if (32768 and dt) <> 0 then {  /*負の数チェック!
1590       dt=dt-65536
1600    }
1610 return(dt)
1620 endfunc
1630 /*
1640 /* 改行コード出力
1650 /*
1660 func cr_put()
1670   fputc(&HD,fp2)
1680   fputc(&HA,fp2)
1690 endfunc
```

▶「スパII」を買ってからRAMを増設しようとする私。なにか違う……。

喜多　清高(25)兵庫県

グラフィックを扱う

Ishigami Tatsuya 石上 達也

従来ビットマップだった部分に対して描画命令が実行可能。これでまたSX-BASICでの処理内容が大きく広がります。新しく追加されたグラフィック命令を探ってみましょう。

読者，関係者各位のおかげをもちまして，SX-BASICもようやくver.0.5をリリースすることができました。ver.0.5にいたり，ようやく方向性のようなものも見えてきました。特に10月号のディスクに収録されたアプリケーションは，私が見ても，えっ，これが，SX-BASIC？　と驚いてしまうようなレベルのものでした（こないだからこればっか）。

特に，郡氏によるゲーム類を見ていると，一刻も早くコンパイラを用意せねばと思うのですが，あとで「しまった」とならないように，いまはじっくり仕様を練っています（特にマルチページ，マルチウィンドウ）。そんなわけで，もうしばらくSX-BASICコンパイラはお待ちください。

BUG BUG BUG

「もみじ狩りPRO-68K」に掲載されたウィンドウデザイナには，「出力ファイルの名前を与えていない状態で，ビットマップアイコンのプロパティ設定を行おうとすると，

『リソースファイル「WIND.LB」がオープンできません』

とダイアログを開くというエラーがありました。54ページのリスト1に示すパッチリストを入力し，wind.xとwind.bfdをカレントディレクトリに置いて，

A＞BUP wind

のように変更を加えてください。

グラフィック処理

前回収録されたver.0.5からは，SX-BASICでもグラフィックを扱うことができるようになりました。まずは，そのあたりから解説していきましょう。

これまでにも，何回か説明してきたようにSX-WINDOWには，

ビットマップデータ

ドローデータ（あるいはgScript）

という2種類のグラフィックを表すデータ形式があります。

前者は，点の集まりとして図形を表す方法，後者は，始点・終点（あるいは，中心・半径など）の座標を数値で表す方法です。

ver.0.3までのSX-BASICはビットマップ形式のグラフィックデータしか扱えませんでしたが，ver.0.5からはドロー形式のデータも扱えるようになりました。

SX-BASICでは，必要に応じてウィンドウ内にアイテムを配置することで，目的のプログラムを作成していきました。テキスト入力用にはテキストアイテム，文字ボタン用には文字ボタンアイテム，といった具合です。ビットマップデータ表示にはビットマップアイテムを用います。

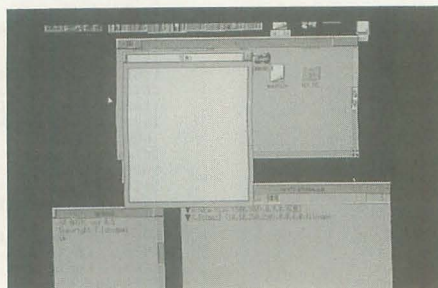
で，ドローデータなのですが，今回，以下の理由により専用のアイテムは用意していません。

1）ビットマップデータとドローデータは同じ領域に併用される場合が多いと考えられる

2）いたずらに，アイテムの種類を増やす，という方向はあまり感心できない

さて，具体的な描画方法ですが，ビットマップアイテムへのメソッド指定で行います。

たとえば，

▼Window Size (300,300), 0,0,0,写真1

というウィンドウ内に，

▼8, Bitmap1, (10,10,290,290), 0,0,0,0,filename

で示されるようなビットマップアイテムがあった場合（写真1。この場合，カレントディレクトリに「filename」という名前のPAT4形式データファイルが必要です），

Bitmap1.forecolor＝11

Bitmap1.penmode＝0

Bitmap1.pensize＝&H00010001

Bitmap1.line＝10,10,30,30

を実行すると，写真2のように直線が描画されます。

写真3はプロッタと呼ばれる図形の出力装置です。通常のプリンタが点の集まりとして図形や文字を出力するのに対し（プリンタの性能を表すDPI＝Dot Per Inchとは，1インチあたり何ドットの点を打てるか，ということを表しています），プロッタは，ペンを実際に動かして紙に図形や文字を描きます。コンピュータからプリンタへ送る命令は，「ここと ここと ここに点を打って，1/180インチ改行して……」という形式ですが，プロッタへの命令は「座標（1000,1200）でペンを紙にくっつけ，くっつけたまま座標（1200,1400）までペンを動かす……」という形式になります。

このプロッタになぞらえて考えると，ドローデータの理解が容易になります。

先ほどのプログラムを順に説明すると，

1）これから扱う図形の描画色を指定し

写真1　ビットマップを開く

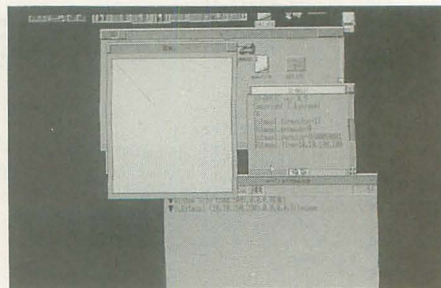



写真2　図形描写後

ます。ここでは，黒を指定しています
2）図形の描画を始めるにあたって，これから使うペンの種類を指定します（ペンモード）。ここでは，インクの色はフォアグラウンドカラー(つまり，描画色＝黒)。インクの種類は普通（pset,後述）
3）ペンの大きさ（太さ）を指定します。上位ワードがX方向の大きさ（この場合は1），下位ワードがY方向の大きさ（同1）を表しています
4）2），3）で指定したペンを用いて，実際に描画を行います

スクリプト(script)

script［C］the written form of a speech play, play or broadcast.
（ロングマンアクティブ学習英英辞典，桐原書店）
だそうです。

最近のゲームには，ビデオモードとかいって，自分のキャラクターがゲーム中どのように動いたか眺めるモードがあります。で，これは，ゲーム中のキャラクターの動きをどこかに記録しておいたからできる技なわけで，このような記録をスクリプトといいます。

SX-BASICには，ビットマップアイテムに対して行ったグラフィック描画をこのようにスクリプト化して記録しておく機能があります。

SX-WINDOWにはグラフィック描画のスクリプトは～にすべし，とすでにシャープによって決められています。この方式に則ったデータをPICTといったりグラフィックスクリプトといったりします（詳しくは，本誌1994年8月号「PICTの活用」を参照）。

戯曲や台本にも短いものや長いものがあるように，このスクリプトデータの長さも可変なので，例によって「メモリハンドル」で扱います。SX-BASICでは，

```
handle=Bitmap1.script
```

のように，scriptプロパティでその値を得ることができます。

逆に，

```
Bitmap1.script=handle
```

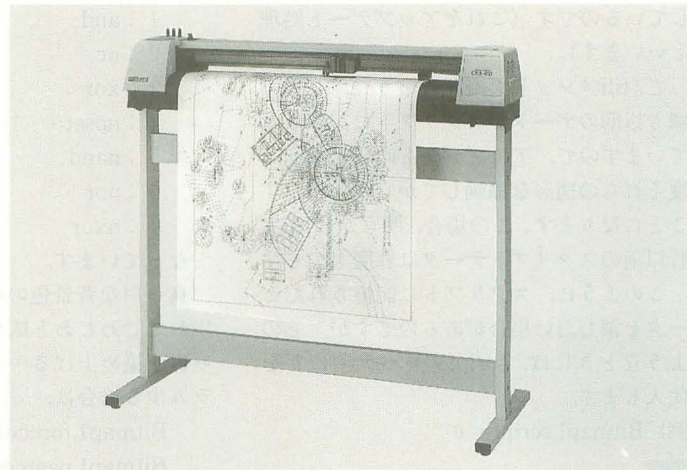
とすることで，新たなデータを指定することもできます。

写真3　プロッタ

少し専門的な話ですが，新しいスクリプトのデータが指定されると，
1）元からあったスクリプトデータは，ハンドルごと消去される（なにもなかった場合はなにもしない）
2）新たにハンドルを作成する
3）与えられたデータを2）のハンドルへコピーする
という手順が踏まれます。なお，SX-BASICの終了時には，これらのデータを収めたハンドルは自動的に破棄されますので，プログラマが，ハンドルの破棄を明示的に行う必要はありません。

つまり，

```
int i, hdl
hdl=Bitmap1.script
for i=0 to 9
  Bitmap2[i].script=hdl
next
```

などのように，じゃんじゃんハンドルを作成していっても構いません。SX-BASICがいらなくなったハンドルを自動的に破棄していきます。

場合によっては，いままで描いてきた図形を消去したいこともあるかもしれません。

X-BASICだと，このような場合，

```
wipe()
```

としてやりました。要は描画範囲の塗りつぶしです。SX-BASICでも，この方法は使えますがあまりスマートなものではありません。

X-BASICでは，途中の経過がどうであれ，画面を塗りつぶせばそれまで描かれた図形は画面上から消去されます。ところが，SX-BASICには上述のスクリプトという機能があり途中の経過もすべて記録されています。

さて，ウィンドウエンジンをほかのウィンドウの下にもう一度手前に持ってくると，隠れる前と変わらない文字・図形などが表示されています。これは，隠れる前の画像データがどこかに保存されているわけではなく，ウィンドウの一部が再び現れたときにはスクリプトデータをもとに本来こうあるべきだという予想をつけて画像を再構成

上位バイト，下位バイト

いまさらいうまでもなく，「0」か「1」かの区別が1ビットの情報量で，これが8つ集まると1バイト，16個集まると1ワード，32個集まると1ロングワードとなります（68000では）。ということは，

2ワードの持つ情報量＝1ロングワードの持つ情報量

です。

X68000は，ロングワードを標準の数値データ形式としています。ですから，なるべくSX-BASICでも扱うデータはロングワード形式にしておいたほうがよさそうです。

必ず1ワードで表現できる（範囲が0～2^{16}=65535）とわかりきっているデータを2つ並べる必要があるときは，2つまとめて1ロングワードで表しておいたほうが標準形式に則っています。

ペンモード，ペンサイズのようなデータがまさにこの場合なのですが，もともとは2つのデータなのですから，場合によっては区別する必要が出てくるかもしれません。そこで，図4の左側にあるデータを上位ワード，右側にあるデータを下位ワードと呼ぶことにします。

上位ワード，下位ワードを示した変数をひとつのロングワード変数で表すには，

```
longword=hiword＊&h10000+loword
```

のようにします。

逆に，ロングワード変数から2つのデータを切り離すときには，

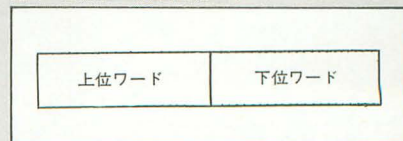
```
hiword=longword/&h10000
loword=longword mod &h10000
```

のようにします。さらに，ビット演算を用いて，

```
hiword=longword shl 16
loword=longword and  &hffff
```

としても同様の結果が得られます。

図4　上位ワードと下位ワード

上位ワード	下位ワード

しているのです（これをアップデート処理といいます）。

で，fillメソッドで綺麗に塗りつぶした領域も以前のデータがスクリプトに記録されていますので，アップデート時にはもう一度それらの図形を描画してから塗りつぶすことになります。この場合，明らかにfill実行以前のスクリプトデータは邪魔ものです。

このように，スクリプトに記憶されたデータを消したい場合があるのですが，そのようなときには，scriptプロパティに0を代入します。

例）Bitmap1.script＝0

ペンモードについて

先ほど，グラフィックの描画に用いるペンの「ペンモード」という項目がありました。このペンモードというのは，

上位ワード
　0：フォアグラウンドカラー(描画色)で描画
　1：バックグラウンドカラー(背景色)で描画

下位ワード
　0：pset
　1：and
　2：or
　3：xor
　4：npset
　5：nand
　6：nor
　7：nxor

となっています。

真っ白な背景色の中に黒い点がポツリと現れてじわじわと広がり，やがて全体を真っ黒く染め上げる……ようなことをプログラムする場合は，

```
Bitmap1.forecolor=11
Bitmap1.penmode=&h00000000
```

を指定し，小さな円，少しだけ小さな円，若干小さな円，心持ち小さな円……を描いていけば実現できます。

例）
```
for i=1 to 10
  Bitmap1.circle=50,50,i
next
```

真っ白な背景の中になにか図形が描かれていて，中心からじわじわと消去されていく，といった場合には，

```
Bitmap1.penmode=&h00010000
```

とし，同様の描画を行えば実現できます。

以上のように，ペンの色を背景色と描画色どちらに指定するかは描く図形が「図」の情報であるか「地」の情報であるかによって使い分けます。

さて，問題は残りの下位バイトです。

写真3のプロッタのイメージでは，ここに0以外の値を当てはめた場合に相当するものはありません。ここに0以外の値を当てはめた場合に実現される機能は二度塗りを行った場合に違いが出てきます。

二度塗りとは，すでになんらかの図形が描かれている場所にもう一度ペンを持っていって図形を描くことです。

このとき，そのまま上書きするのが0の「pset」です。そのほかは図1を参照してください。

詳しい説明は省きますが，

　and：Both A and B
　or：Either A or B
　xor：exclusive-or（排他的論理和）

の意です。

xorはちょっとわかりづらいのですが，「or」の親戚です。唯一の違いは，A,Bともに黒の場合は，白となってしまうことです。接頭語の「n」はnotの略です。出力が反転（白→黒，黒→白）ます。

ということで，あとは類推してください。

図1　ペンモード（SX本P79図25より転載）

グラフィックの座標系について

まず，グラフィックを表示するには表示する領域が必要です。X-BASICの場合は，グラフィック領域の好きなところに描画を行えました。これは，X-BASICが基本的に単独のプログラムを実行する環境だから実現された機能です。SX-BASICは同時に複数のプログラムを実行できるようにしておかなければなりませんので，プログラムごとにグラフィックを描画する領域を分けて，互いに干渉しないようにしなければなりません。

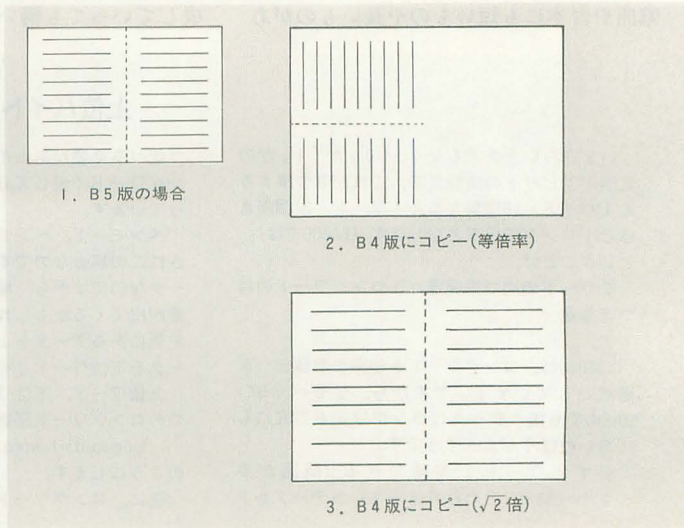
SX-WINDOWのアプリケーションは基本的に自分のウィンドウを持っていて，各種の情報はそのウィンドウ内に行うことになっています。グラフィック描画も例外ではなく，「どのような」図形を描画するか指定する命令は「どこの」ウィンドウに描画するのか指定する命令と対をなすことになっています。

図2　グローバル座標とローカル座標とアイテム内ローカル座標

図3　データコピーのいろいろ

1．B5版の場合

2．B4版にコピー(等倍率)

3．B4版にコピー($\sqrt{2}$倍)

具体的にいうと，SX-WINDOWはウィンドウ内にディスプレイ画面とはまったく別の座標系を持っています（以下，ローカル座標。ディスプレイの座標はグローバル座標)。こうしておけば，すべての座標はウィンドウの左上部分を原点としたローカル座標で表されていますから，ウィンドウをディスプレイ上のどこへ移動しようと，いつでも，座標データを変更することなしに，同じ図形を描画できるのです。

さて，同じことがグラフィック描画にもいえます。確かに，ウィンドウ内の位置を表すローカル座標ですべての座標を統一しておけばシンプルなデータ管理が行えるかもしれません。

しかし，編集部内で作成されたプログラムや投稿されたプログラムを見ると，ビットマップアイテムの移動は禁止しないほうがよさそうです（花札，トランプカードの移動など）。

で，ビットマップアイテムを移動するたびに，その中に描く図形の座標系も変更するというのは面倒です。先ほどのローカル座標とグローバル座標の概念が思い出されます。

そんなわけで，グラフィック描画にはウィンドウ内のローカル座標に対して，さらにローカルな座標系が用意されています(図2)。

もうひとつの理由はビットマップアイテム内に描画するグラフィックはクリプト形式（ドロー形式）です。ということは，いつかもお話ししたように，データに可搬性を持たせることができるということです。

SX-BASICでも，

clipbord.pict＝bitmap1.script

のように，ビットマップアイテム内に描かれたグラフィックをクリップボードに転送することもできます。

このように，データが元の場所を離れてさまざまなところへ転送されてくるようになると，元の場所に固有な座標系を用いるわけにはいきません。このことも新たに座標系を導入した一因となっています。

resizeプロパティ

たいていのウィンドウシステムはウィンドウの大きさを変える機能があります。大事な情報を表示しているウィンドウは大きく広げ，そうでないウィンドウはそれなりの大きさにしておき，限りあるデスクトップ画面を有効に活用します。

ウィンドウを大きくする場合，目的は2通り考えられます。

1）エディタのように文字の大きさを変えないで，一度に見ることのできる情報量を上げたい場合

2）大事な情報なのでもっと拡大して見たい場合

MS-WINDOWSにはウィンドウのタイトルバーに「最大化」「最小化」を指定するボタンがついています。最初の頃はこれで結構遊んでいたのですが，あるときカレンダーをいじっていてびっくりしました。「最大化」ボタンを押すと，確かにウィンドウは画面いっぱいに広がるのですが，そこに表示されているのは，そうでない場合とまったく変わらない情報でウィンドウが広がった部分は空白で埋め尽くされていたのです(写真4)。ちょっと考えてみればまったく当たり前のことなのですが，初めに見たときはびっくりしました。普通，ボタンなども一緒に拡大されると思うじゃないですか。

この例と同様に，グラフィックデータをあっちこっちへ移動しているとデータの意味あいが若干ずれてくることがあります。小さな領域のビットマップアイテム内に描かれたグラフィックデータを大きな領域を持つビットマップアイテムへコピーした場合，

1）そのままの大きさで表示

2）ビットマップアイテムの大きさにあわせて，拡大・縮小を行う

の2通りの表示方法が考えられます。

コピー機の例で考えてみましょう。

図3.1のような図形がB5判の用紙に描かれていたとしましょう。これは用紙を半分に分ける直線です。左がアンケートの問題で右が回答用紙という感じです。

で，B5判の紙がたまたま切れていたのでB4判の紙にコピーする場合を考えましょう。原版をそのままコピー機にかけたのでは，図3.2のようになってしまいます。ちょっと気のきいたコピー機なら「用紙サイズ確認」とかいって注意を促してくるかもしれません。

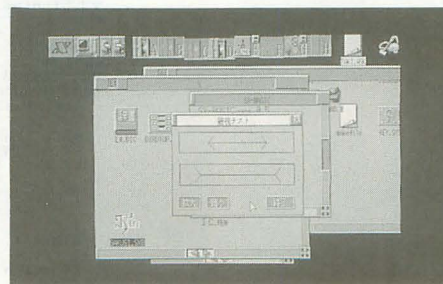



写真4　錯視ゲーム

で，倍率144％（＝$\sqrt{2}$）にセットすれば，図3.3のような結果が得られるわけです。普通はこれで問題ないのですが，右半分が返送用の葉書になっていたりすると，やはり図3.2のほうが正解となったりします。

このように，ある図形を大きさの違う領域にコピーする際，その方法を区別しなければならないのですが，この区別を行うのがresizeプロパティです。

resize＝0のときはビットマップアイテムは渡されたグラフィックスクリプトをそのままの状態で表示し，resize≠0のときは適宜，伸長/縮小処理を行ってから表示します。

アイテム追加のフォーマット変更

ver.0.5になり，いくつかのアイテム配置に関するフォーマットが変更されました。

▼Window Size (x,y), id,growbox, drag,caption

idの項目が追加されました。そのほかは，以前のバージョンとまったく同じです。

idはツールバーのリソースidです。ツールバーを使用しない場合は，0を指定してください。

ツールバーは，各種リソースの複合体として実現されています。詳しくは，「開発キットツール集」のマニュアル，ドキュメントを参照してください。

注）この機能を使用するには，シャープ製「開発キットツール集」に収録されている，リソースタイプ「dmdf」で示されるコードリソースが必要です。

▼11, Name, (x1,y1,x2,y2), isarray, index,id

11：

以下に続くデータがアコーディオンメニューアイテムであることを示します。変更してはいけません。

Name (str)：

アイテムの名前。

isarray：

このアイテムが配列化されているかを示すフラグ。0で非配列アイテム，それ以外で配列アイテムです。

index：

配列アイテムの配列番号，非配列アイテムの場合，意味はありません。

id：

アコーディオンメニューの内容を収めたリソースIDです。アコーディオンメニューは，各種リソースの複合体として実現されています。詳しくは「開発キットツール集」

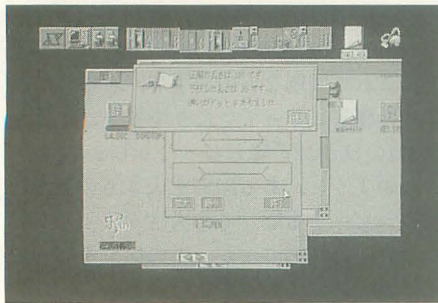



写真5　錯視ゲーム（結果表示）

写真6　錯視ゲーム（scriptを消去しないバージョン）

のマニュアル，ドキュメントを参照してください。

注）この機能を使用するには，シャープ製「開発キットツール集」に収録されている，リソースタイプ「dmdf」で示されるコードリソースが必要です。

サンプルの遊び方

いきなり問題。

>————<

と，

<————>

どちらが，長いでしょうか。

正解）下の直線のほうが1ミリほど長いように写植してあります。

とまあ，人間の目というのは，かようにいい加減なものでありまして，そのいい加減具合を計量化するプログラムを作ってみました。

アイデアは前回の百人一首と同様，ワープロソフトの「一太郎」から拝借しました。

リスト1のプログラムをエディタなどを用いて，入力し，ファイルに保存してください。次にSX-WINDOW上からそのファイルアイコンをSX-BASICへ放り込みます。入力ミスがなければ写真4のような画面が表れるはずです。

先ほどのような，人を欺くための直線が2本表示されています。左下の「拡大」「縮小」と書かれた文字ボタン（標準ボタン）を操作し，下の直線の長さを調節してみます。いろいろ試して，2本の直線の長さが同じになったと思ったら，右下隅にある「終了」ボタンを押します。

すると，ダイアログが開いて，実際のところ，どうなっているのかを教えてくれます（写真5）。

サンプルプログラムについて

特に変わったことはしていません。プログラム全域を通じて使っている変数も，

l1：

正解の直線の長さ

l2：

解答の直線の長さ

だけです。

「拡大」「縮小」ボタンが押された場合，解答の直線の長さを，4ドット刻みで変化させています。1ドット刻みだと，やっていてイライラしますし，あまり大きく刻みすぎると，問題が簡単になってしまいます。いろいろ実験をしてみて，4という値に落ち着きました。

ですから，l1−l2は必ず4の倍数にならなければなりません。回答者は4ドットずつしか長さを調節できませんから，

l1＝112，l2＝113

だった場合，

「拡大」→　l1＝112，l2＝117

「縮小」→　l1＝112，l2＝109

となって，永遠に正しい値にたどり着くことができなくなってしまいます。

プログラム，7,8行目の，

l1＝（rand（）　and 31）＊4＋70

l2＝（rand（）　and 31）＊4＋70

というのはそのためのものです。こうしておけばl1−l2は必ず4の倍数になりますのでそのような心配はなくなります（rand（）and 31で0～31の値を得，それを必ず4倍してから70を足すので……というわけで続きは各自で考えてください）。

グラフィックですが，問題の直線はアイテム「Bitmap1」に，解答の直線はアイテム「Bitmap2」にそれぞれ描画しています。実際に描画を行っているのは関数DrawL1（），DrawL2（）です。

両アイテムとも横208ドット，縦32ドットの大きさですから，その中心座標は（104,16）です。それを踏まえて各直線の長さに応じてx1,x2を計算し，直線を描いています。

描画を始めるにあたって，

Bitmap1.script＝0

のように，初めにグラフィックスクリプトの記録を消去しています。これを行わないと前回までの描画が残されてしまいます。試しに17行と31行をコメント行にして，プログラムを実行すると，この機能の役割がよくわかるのではないかと思います（写真6）。

リスト

```
▼Window Size (237,133),0,0,0,錯視テスト
/* ここで、初期化に必要な処理を行なって下さい
int l1,l2
Init()
end
func Init()
  l1 = (rand() and 31) * 4 + 70
  l2 = (rand() and 31) * 4 + 70
  DrawL1()
  DrawL2()
endfunc
/* 正解の直線を描く
func DrawL1()
  int x1,x2
  x1 = 104 - l1 / 2
  x2 = 104 + l1 / 2
  Bitmap1.script = 0
  Bitmap1.forecolor = 11
  Bitmap1.penmode = 0
  Bitmap1.line = x1,16,x2,16
  Bitmap1.line = x1,16,x1+10,6
  Bitmap1.line = x1,16,x1+10,26
  Bitmap1.line = x2,16,x2-10,6
  Bitmap1.line = x2,16,x2-10,26
endfunc
/* 回答の直線を描く
func DrawL2()
  int x1,x2
  x1 = 104 - l2 / 2
  x2 = 104 + l2 / 2
  Bitmap2.script = 0
  Bitmap2.forecolor = 11
  Bitmap2.penmode = 0
  Bitmap2.line = x1,16,x2,16
  Bitmap2.line = x1,16,x1-10,6
  Bitmap2.line = x1,16,x1-10,26
  Bitmap2.line = x2,16,x2+10,6
  Bitmap2.line = x2,16,x2+10,26
endfunc
▼8,Bitmap1 (12,8,220,40),0,0,0,0,
▼8,Bitmap2 (12,56,220,88),0,0,0,0,
▼3,StnBtn1 (10,104,50,124),0,0,拡大
func StnBtn1_Click()
  l2 = l2 + 4
  DrawL2()
endfunc
▼3,StnBtn2 (60,104,100,124),0,0,縮小
func StnBtn2_Click()
  l2 = l2 - 4
  DrawL2()
endfunc
▼3,StnBtn3 (168,104,220,124),0,0,終了
func StnBtn3_Click()
  str s1,s2,s3
  int ret
  s1 = "正解の長さは"+itoa(l1)+"です"+chr$(13)
  s2 = "予想した長さは"+itoa(l2)+"です"+chr$(13)
  if l1 = l2 then {
      s3 = "正解です！！"
  } else {
      s3 = "違いがドット"+itoa(abs(l1-l2))+"ありました"
  }
  alart(1, s1+s2+s3)
/*  ret = alart(4, "もう一度、挑戦しますか？")
  if ret = 0 then  Init() else end
endfunc
```

〈対応機種一覧〉●MZ-80 K/C/700/1500 ●MZ-80 B/2000 ●MZ-2500/2861 ●X1 ●X1 turbo/Z ●PC-8001/8801/88 ●SMC-777/C ●PASOPIA/5 ●PASOPIA/7 ●FM-7/77/AV ●MSX/2/2+/turbo R ●PC-286/386/486/9801/98/9821 ●X68000/X68030
掲載されたプログラムの利用には各機種用のS-OS "SWORD" システムが必要です。

第151部　B-GALETS2

●S-OSのゲーム作法

制限のあるS-OSの世界では，その制限からほとんど暗黙の了解っぽい作法がまかりとおっています。まかりとおっている，というい方は，ちょっと悪いイメージなので「先人たちの作法に従っている」とでもいったらいいのでしょうか。

ちなみにゲームでは，好きなときにSHIFT+BREAKキーでS-OSモニタに戻ってこれるようにするとか，コンパイラなどではQキー(QUIT)で終了とか。いくつかアプリケーションを使ったことのある人なら，思い当たることがあるでしょう。

また，S-OSではリアルタイムキー入力はできても同時キー入力ができません。そうなると，弾を撃ちながら移動を行わなくてはならないシューティングゲームでは，非常に困ってしまいます。なにしろ，迫りくる敵の弾を避けつつ，迎撃しなくてはならないのです。シューティングゲームでは，移動と弾を撃つ動作を同時に行う必要がありますからね。弾を撃つたびに自機の移動が止まってしまうのでは，ゲームにならないでしょう。

そこで，いままでに発表されたシューティングゲームでは，ほとんどすべて弾の発射をトグルスイッチにしています。要するに，いったん弾の発射キーを押したら出っぱなし（連射状態）になり，もう一度弾の発射キーが押されるとストップする，というようなシステムとなっているのです。こうすれば，プレイヤーは移動に専念でき，快適にゲームを楽しむことができます。

今回発表した「B-GALET2」も，投稿されたものは弾を発射している間の移動ができないようになっていました。一応，投稿原稿には「S-OSでは同時キー入力ができないから……」と書かれていたので，とりあえず遊んでみると……やはりストレスが溜まる溜まる(笑)。こりゃあかん，てなわけで作者の方には申しわけありませんが，連射仕様にプログラムを変更させていただきました。このためゲームバランスが崩れてしまうかな？　と心配しましたが，遊びやすくなったぶん，ほどよい難易度になりました（ゲームスピードを速めにしておくとなおよい）。

そうそう，もうひとつゲームの操作における基本事項があります。それは，プレイヤーが操作するキャラクターで上下左右の移動を行う場合は，テンキーの8,2,4,6キーではなく，Kキーを中心としたI,M,J,Lキーを基本とするものです。要するにテンキーのない機種のために，メインキーで操作できる環境もサポートしてほしい，ということです。「B-GALET2」では，自機の左右移動が4,6キーとなっていますが，以下のアドレスを変更することでJ,Lキーで左右の移動ができるようになります。

A135　34→4A

A139　36→4C

それにしても単純な割には難しいゲームです（X68000 XVIで遊んでいるのが悪いという話も)。まだそんなに遊んでいないのですが，いまだに1LOOPすらクリアできないのはなんとなく悔しいですね。時間があったらもう一度挑戦してみますか。

なお，ページの都合によりソースリストの掲載は見合わさせていただきました。

●MOOK化計画，次なるステップへ（2）

10月号のこのTHE SENTINELで募集を行った「マニュアル作成スタッフ」。協力のハガキがてんこ盛り，とまではいきませんが，なんとか十分な人数が集まりつつあります。とりあえず，現在の段階では必要十分な人数が集まった，ということで募集を打ち切らせていただきます。

すでに協力スタッフとして登録された方々には，それぞれの作業内容が記された封書が届いていることでしょう。あまり無理をせず，マイペースで作業を進めてください（さすがに1カ月以上連絡がこないと不安になるだろうけど)。

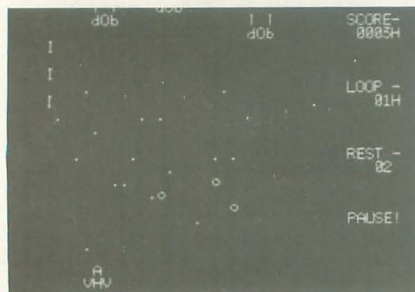
では，また来月号のTHE SENTINELでお会いしましょう。

1994■インデックス

全機種共通
S-OS"SWORD"要

B-GALETS2

Yoshida Masayuki
吉田 昌之

敵の動きが気持ちいい，ひたすら弾を避けながら敵を撃つだけのとってもシンプルなシューティングゲームです。反射神経を駆使して，制作者も到達したことのない最終レベルを目指しましょう。

S-OS"SWORD"用にシューティングゲームが完成しましたので，ここに発表します。といっても，パワーアップやデカキャラなど，凝ったものはなにもありません。

手抜きといわれてしまえばそうなのですか，私自身の技術力の問題もあって，これで限界なのです。

しかし，その制限の中で，可能な限りのアイデアを詰め込んだつもりです。敵の攻撃をかわし，攻撃する。いいようによっては，純粋なシューティングといえるかもしれません。

遊び方

ゲーム内容を言葉で表現すると，操作はギャラクシアン風で，敵の攻撃はスターフォースみたいな感じといったところでしょうか。まず，「MACINTO-C」などのマシン語入力ツールを使いダンプリストを入力してから，

#S B-GALETS2：A000：AE0E：A000

でいったんセーブしましょう。セーブが終わったら，

#JA000

で実行できます。すると実にシンプルなタイトル画面になります。スペースキーを押すとゲームがスタートし，「Q」キーで終了します。このとき「H」キーでハードモード，「E」キーでイージーモードになります。「S」キーで，スピード調整もできます。「S」キーを押したあとにA(速い)～Z(遅い)のキーで26段階に設定可能です。ゲームスピードの調整は，背景の星の数を増減することによって行っています。

ただし，X68030で実行した場合，最も遅くしてもまだ速すぎる状態です。30SPEED.Rなどを使って，マシンのスピード自体を遅くしないと，ゲームにならなくなってしまいます。注意してください。

ゲーム中は「4」と「6」で左右移動。上下移動はありません。スペースキーを押すことで弾を3発まで連射できます。一度，スペースキーを押すと弾は連射され，もう一度押すと弾は発射されません。あと，パワーアップの類は一切存在しません。ただ，ひたすら敵を迎撃するだけです。

ゲーム内容について

ゲームに登場するたいていの敵は，同じキャラクターで2種類の攻撃パターンをもっています。なかには，どさくさに紛れて1匹だけ種類が違うという，ほとんどだましうちのような攻撃もありますので注意しましょう。16種類のキャラクターが登場し，中にはワープして出現するものもいます。

敵の出現パターンが全部出尽くすと，LOOPカウントがアップして，難易度が上がり，自機が1機増えます。難易度は，確か5周目で最高になるはずです（作った本人もそこまでいったことがないのです）。

プログラムについて

一般的に「遅い」とされる，IX，IYレジスタをこれでもかっ，というほど使っています。

しかし，この程度のプログラムでは，それほどネックにはならなかったようです(少なくともX1，X68030では，十分な速さで動くことを確認しています)。逆に，これらを使うことで「まず確実に動くプログラム」が楽に作れました。

何カ所かは，IX，IYレジスタを使わないように最適化していますが，大半がそのままです。自機の移動には，特別なことをなにもしていません。

敵キャラの移動は，共通ルーチンを作成し，移動方向を示す数字を並べるだけで，動かせるようになっています。

また，単純な移動のほかに，分岐，繰り返し，誘導移動，表示パターン変更などの機能があり，工夫ひとつでいろいろなタイプが作れます。

特に，個性を出すための処理だけは，専用にプログラムしてやります。今回は，出現パターンについてのみ，個別に用意したものがあります。この方法のよいところは，一度完成してしまえば，新キャラの作成が容易であるということです。

敵の攻撃パターンは「どの敵が何匹」というデータを並べて作っています。

しかし，出現位置はほとんど乱数ですから，「パターンゲームのようでいて，乱数要素も強い」という状態になっています。

背景の星は，あとからつけ足したものですが，表示にちょっとひねりを加えて，ほかのキャラクター表示に干渉しないようになっています。

さらに，自機の撃つ弾は，絶対に自機とは接触しません。

よって，自機と重なる可能性のあるものは，「敵」と「敵の弾」だけという状態になります。この理論（？）により，自機の死亡判定が「自機の表示が崩れたら接触した」という，実に手抜きな処理になっています。

しかし，これにはいいところもあって，自機の当たり判定が厳密にできるようになっています。というのは，自機のキャラの右上と左上の空白部分は，ちゃんと当たり判定がないのです。だから，ギリギリの避けも可能です（狙ってできるものではありませんが）。逆に，敵側は，単純に座標の差分を計算して行っているので，空白部分でも当たります。

乱数は，処理の軽い方法として「N×5＋1」という計算を繰り返すことによって発生させています。

これに，そのとき押されているキーによって，計算回数を変化するようにし，操作している人も乱数の要素としています。

終わりに

とにかく「完成のためなら可能な限り省略してしまえ」と作ってしまいました。最初にこのゲームが動いたときには，敵が5種類程度のものだったのですが，ちょっと改良しては投げ出してを繰り返すうちに3年もたってしまいました。

3年もかかってこの程度，というのも情けないものがありますが，ひとつ完成させたという実績は，これからプログラミングをしていくうえで，きっとプラスになってくれると思っています。

次はYGCSシステムにも挑戦してみたいのですが，時間の問題もあって，ダンプの入力すら手をつけられない状態です。

次に作ろうとしているものが，はたしていつ完成するかはわかりませんが，やる気はあるのでこれからもがんばっていくつもりです。

リスト

```
A000 3E 28 CD 30 20 21 10 00 : B4
A008 22 D7 AC 97 32 2A A1 3E : 77
A010 0C CD F4 1F 21 0B 06 CD : EB
A018 1E 20 CD E2 1F 42 20 2D : 9B
A020 20 47 20 41 20 4C 20 45 : 99
A028 20 54 20 53 20 5D 5B 00 : BF
A030 21 0B 0B CD 1E 20 CD E2 : F1
A038 1F 50 75 73 68 20 20 5B : 5A
A040 53 50 41 43 45 5D 20 20 : 09
A048 6B 65 79 00 21 0D 11 CD : 55
A050 1E 20 CD E2 1F 48 49 2D : CA
A058 53 43 4F 52 45 3A 20 00 : D6
A060 2A D7 AC CD BE 1F 3E 48 : DD
A068 CD F4 1F CD D0 1F FE 51 : EB
A070 C8 FE 48 CA D8 A6 FE 45 : 99
A078 CA 09 A7 FE 53 CA 39 A7 : 75
-----------------------------------
SUM: C2 CC 8A 75 DB 1B 4C 59 FAFE

A080 FE 20 20 E7 CD 32 A6 CD : 97
A088 D8 A5 06 03 21 0E 0A CD : 8C
A090 1E 20 CD E2 1F 53 54 41 : F4
A098 52 54 00 CD B8 A5 21 0E : FF
A0A0 0A CD 1E 20 CD E2 1F 20 : 03
A0A8 20 20 20 20 00 CD B8 A5 : AA
A0B0 10 DA 21 11 17 3E 48 CD : 86
A0B8 1E 20 CD F4 1F 06 06 2E : 58
A0C0 11 3E 16 90 67 CD 1E 20 : 67
A0C8 CD E2 1F 20 1F 1D 41 00 : 6B
A0D0 26 17 3E 10 90 6F CD 1E : 75
A0D8 20 CD E2 1F 20 56 00 3E : A2
A0E0 11 80 6F CD 1E 20 CD E2 : BA
A0E8 1F 56 20 00 CD B3 A5 10 : CA
A0F0 CE CD 00 A1 CD ED A4 C2 : 5C
A0F8 0D A5 CD 7D A3 C3 F1 A0 : F3
-----------------------------------
SUM: CD 6C D0 A8 59 5D 7D 79 5126

A100 CD 2B A1 CD EF A5 CD BD : 84
A108 A1 CD 86 A4 CD BD A1 CD : 90
A110 86 A4 CD F4 A1 CD 44 A3 : 40
A118 CD EF A5 C9 FE 20 20 09 : 71
A120 F5 3A 2A A1 2F 32 2A A1 : 26
A128 F1 C9 00 2A DD AC CD D0 : 0A
A130 1F CD 1C A1 FE 34 28 2A : 2D
A138 FE 36 28 32 FE 50 CC 81 : 29
A140 A7 F5 3A 2A A1 B7 20 32 : AA
A148 F1 22 DD AC CD 1E 20 CD : 74
A150 E2 1F 20 41 1F 1D 1D 56 : 11
A158 48 56 00 C9 2A DD AC C3 : DD
A160 4C A1 CD AD A1 3E 00 BD : 03
A168 28 D7 2D C3 41 A1 CD AD : 4B
A170 A1 3E 1F BD 28 CB 2C C3 : 9D
A178 41 A1 E5 CD 9C A5 21 66 : 5C
-----------------------------------
SUM: DC 74 3C A6 C0 CF E0 FD 8A82

A180 AD 06 03 7E B7 28 07 23 : 3D
A188 23 23 10 F7 18 1B 3E 01 : BF
A190 77 3A DD AC 23 3C 77 3A : 4A
A198 DE AC 23 3D 77 57 2B 5E : 41
A1A0 EB CD 1E 20 3E 49 CD F4 : 3E
A1A8 1F E1 C3 48 A1 CD 1E 20 : B7
A1B0 CD E2 1F 20 20 1F 1D 1D : 67
A1B8 20 20 20 00 C9 21 66 AD : 5D
A1C0 06 03 7E 23 B7 28 1E 5E : 05
A1C8 23 56 EB CD 1E 20 3E 20 : CD
A1D0 CD F4 1F 3E 00 BC 28 12 : 14
A1D8 25 CD 1E 20 3E 49 CD F4 : 78
A1E0 1F EB 72 2B 73 23 23 10 : 70
A1E8 D9 C9 EB 2B 2B 3E 00 77 : 98
A1F0 23 C3 E5 A1 DD 21 E6 AC : FC
A1F8 3A E3 AC 47 C5 CD 09 A2 : 4D
-----------------------------------
SUM: 8C 33 C7 72 84 C8 B8 F3 EADC

A200 11 08 00 DD 19 C1 10 F4 : D4
A208 C9 DD 7E 00 B7 C8 DD 6E : EE
A210 02 DD 66 03 DD 5E 04 DD : 64
A218 56 05 CB 7F C2 F9 A2 1A : 1C
A220 13 DD 73 04 DD 72 05 FE : B9
A228 0A 28 28 FE 0B CA 13 A3 : E3
A230 FE 0C CA 5D A2 FE 0E CA : A9
A238 4B A2 FE 0D CC 69 A2 CD : 9C
A240 78 A2 DD 75 02 DD 74 03 : C2
A248 C3 E2 A2 1A DD 77 01 13 : C9
A250 C3 1F A2 EB 4E 23 46 C5 : EB
A258 E1 EB C3 1F A2 CD AC A5 : 6E
A260 E6 01 28 EF 13 13 C3 1F : 06
A268 A2 E5 21 DD AC DD 7E 02 : 8E
A270 BE E1 3E 03 D8 3E 01 C9 : C0
A278 47 CD 1E 20 CD E2 1F 20 : 40
-----------------------------------
SUM: 04 9C 9B 53 F8 D7 23 1B 12B5

A280 20 20 1F 1D 1D 1D 20 20 : F6
A288 20 00 05 28 46 05 28 1C : DC
A290 05 28 34 05 28 1D 05 28 : D8
A298 49 05 28 1E 05 28 2E 05 : F4
A2A0 28 03 05 28 1C 25 3E 00 : D7
A2A8 BC 28 2E C9 24 3E 16 BC : 0F
A2B0 28 27 C9 2D 3E 00 BD 28 : 68
A2B8 20 C9 2C 3E 1F BD 28 19 : 70
A2C0 C9 CD A5 A2 C3 BA A2 CD : C9
A2C8 AC A2 C3 BA A2 CD A5 A2 : 81
A2D0 C3 B3 A2 CD AC A2 C3 B3 : A9
A2D8 A2 3E 00 DD 77 00 DD 77 : 88
A2E0 01 C9 DD 7E 01 E5 87 26 : B8
A2E8 00 6F 11 A8 A7 19 5E 23 : 69
A2F0 56 E1 CD 1E 20 CD E5 1F : 13
A2F8 C9 CD 1E 20 CD E2 1F 20 : C2
-----------------------------------
SUM: B4 AE 8B 2E 4A 5D 84 87 E4A7

A300 20 20 1F 1D 1D 1D 20 20 : F6
A308 20 00 DD 34 00 DD 34 01 : 43
A310 C3 E2 A2 21 6F AD 3A E4 : A2
A318 AC 47 7E B7 28 08 23 23 : 9E
A320 23 10 F7 C3 0E A2 3E 01 : DC
A328 77 DD 7E 02 23 3C 77 DD : 87
A330 7E 03 23 3C 77 57 2B 5E : 37
A338 EB CD 1E 20 3E 6F CD F4 : 64
A340 1F C3 0E A2 21 6F AD 3A : 09
A348 E4 AC 47 7E 23 B7 28 1E : 75
A350 5E 23 56 EB CD 1E 20 3E : 0B
A358 20 CD F4 1F 3E 17 BC 28 : 39
A360 12 24 CD 1E 20 3E 6F CD : BB
A368 F4 1F EB 72 2B 73 23 23 : 54
A370 10 D9 C9 EB 2B 2B 3E 00 : 31
A378 77 23 C3 6E A3 21 E6 AC : 21
-----------------------------------
SUM: C0 A4 B5 5D 02 AB C5 B2 3D80

A380 11 08 00 3A E3 AC 47 7E : A7
A388 B7 CA 90 A3 19 10 F8 C9 : 9E
A390 3A E2 AC B7 CA B2 A3 E5 : 83
A398 21 07 A9 11 05 00 3D CA : EE
A3A0 A6 A3 19 C3 9E A3 4E 23 : D7
A3A8 46 23 5E 23 56 23 7E E1 : C2
A3B0 C5 C9 21 E1 AC 35 C0 2A : 5B
A3B8 DF AC 7E CB 7F C2 CD A3 : 85
A3C0 32 E2 AC 23 7E 32 E1 AC : 20
A3C8 23 22 DF AC C9 3A E3 AC : 62
A3D0 FE 10 28 04 3C 32 E3 AC : 37
A3D8 3A E4 AC FE 20 28 05 3C : 51
A3E0 3C 32 E4 AC 3A DB AC 3C : FB
A3E8 32 DB AC 3A DC AC 3C 32 : E9
A3F0 DC AC CD B4 A6 21 D4 AB : 4F
A3F8 22 DF AC C3 BA A3 77 23 : 67
-----------------------------------
SUM: AC 86 63 65 03 3C 57 43 E00F

A400 77 CD AC A5 E6 01 CA 11 : 57
A408 A4 3E 00 11 DF A9 C3 16 : 54
A410 A4 3E 1F 11 C7 A9 23 77 : 1C
A418 23 CD AC A5 E6 0F 77 C3 : 70
A420 70 A4 77 23 77 23 CD AC : C1
A428 A5 E6 01 CA 3B A4 CD AC : AE
A430 A5 E6 03 C6 1B 11 59 AA : 83
A438 C3 44 A4 CD AC A5 E6 03 : B2
A440 3C 11 80 AA 77 C3 6C A4 : C1
A448 77 23 77 23 3E 10 77 C3 : BC
A450 6C A4 77 23 77 23 CD 77 : 88
A458 A4 77 CD AC A5 E6 07 23 : 49
A460 77 C3 70 A4 77 23 77 23 : 82
A468 CD 77 A4 77 3E 00 23 77 : 37
A470 23 73 23 72 C3 B2 A3 CD : 10
A478 AC A5 E6 07 47 CD AC A5 : A3
-----------------------------------
SUM: 35 6B EE 1C 7B 5D A0 73 ECBE

A480 E6 0F 80 C6 05 C9 11 08 : 22
A488 00 0E 03 21 66 AD 7E 23 : E6
A490 B7 28 21 3A E3 AC 47 DD : ED
A498 21 E6 AC DD 7E 00 B7 28 : ED
A4A0 0F CB 7F 20 0B 23 7E DD : 02
A4A8 96 03 28 0F 3D 28 0C 2B : 6C
A4B0 DD 19 10 E7 23 23 0D C2 : 02
A4B8 8E A4 C9 2B 7E DD 96 02 : 19
A4C0 28 09 3D 28 06 3D 28 03 : 04
A4C8 C3 B0 A4 CD E0 A4 3E FC : A2
A4D0 DD 77 00 3E 06 DD 77 01 : ED
A4D8 3E 00 2B 77 23 C3 B0 A4 : 1A
A4E0 E5 2A D9 AC 23 22 D9 AC : 5E
A4E8 CD CB A6 E1 C9 2A DD AC : 9B
A4F0 2C CD 1B 20 FE 41 C0 24 : 57
A4F8 CD 1B 20 FE 48 C0 2D CD : 08
-----------------------------------
SUM: 7F C3 96 94 F6 3B EA E9 519D

A500 1B 20 FE 56 C0 2C 2C CD : 74
A508 1B 20 FE 56 C9 2A DD AC : 0B
A510 3E 07 CD E5 A2 CD 03 A1 : 0A
A518 2A DD AC 3E 08 CD E5 A2 : 4D
A520 CD 03 A1 2A DD AC 3E 09 : 6B
A528 CD E5 A2 CD 03 A1 2A DD : CC
A530 AC 3E 0A CD E5 A2 CD 03 : 18
A538 A1 06 20 C5 CD 03 A1 2A : 27
A540 DD AC 3E 0A CD E5 A2 CD : F2
A548 C7 A5 C1 10 EE 3A DC AC : ED
A550 B7 CA 61 A5 3D 32 DC AC : 7E
A558 CD CE A5 CD 5B A6 C3 87 : 58
A560 A0 06 C8 C5 CD 03 A1 CD : 71
A568 7D A3 21 0B 0A CD 1E 20 : 61
A570 CD E2 1F 47 41 4D 45 20 : 08
A578 4F 56 45 52 00 CD C7 A5 : 75
-----------------------------------
SUM: E6 1A 34 4D 30 C3 AF 2D B87E

A580 C1 10 E0 CD CE A5 2A D9 : F4
A588 AC ED 5B D7 AC B7 ED 52 : 6D
A590 DA 0B A0 2A D9 AC 22 D7 : 2D
A598 AC C3 0B A0 08 C5 3A E5 : 06
A5A0 AC 47 87 87 80 3C 32 E5 : D4
A5A8 AC C1 08 C9 CD 9C A5 3A : 86
A5B0 E5 AC C9 C5 06 0F 18 03 : 4F
A5B8 C5 06 32 C5 CD EF A5 C1 : E4
A5C0 CD C7 A5 10 F6 C1 C9 C5 : 8E
A5C8 06 FF 10 FE C1 C9 C5 06 : 68
A5D0 FF CD C7 A5 10 FB C1 C9 : CD
A5D8 3A D6 AC 47 21 CF AD CD : 6D
A5E0 AC A5 E6 1F 77 23 ED 5F : 3C
A5E8 E6 0F 77 23 10 F1 C9 3A : 93
A5F0 D6 AC 47 11 CF AD 1A 6F : DF
A5F8 13 1A 67 CD 1B 20 FE 2E : C8
-----------------------------------
SUM: 7C 68 A3 62 D4 D8 D1 61 A3A3
```

```
A600 20 08 3E 20 CD 1E 20 CD : 5E
A608 F4 1F 24 3E 17 BC 28 15 : 85
A610 CD 1B 20 FE 20 20 08 3E : 8C
A618 2E CD 1E 20 CD F4 1F 7C : 95
A620 12 13 10 D2 C9 1B CD AC : 64
A628 A5 E6 1F 12 13 26 00 C3 : B8
A630 1F A6 21 00 00 22 D9 AC : 8D
A638 3E 01 32 DB AC 3E 02 32 : 6A
A640 DC AC 21 D4 AB 22 DF AC : D5
A648 3E 00 32 E2 AC 3E 0F 32 : 7D
A650 E1 AC 3E 03 32 E3 AC 3C : CB
A658 32 E4 AC 21 10 16 22 DD : 08
A660 AC 21 E6 AC 11 E7 AC 01 : 04
A668 FF 00 36 00 ED B0 3E 0C : 1C
A670 CD F4 1F 21 22 03 CD 1E : 11
A678 20 CD E2 1F 53 43 4F 52 : 25
----------------------------------
SUM: E8 CD 7C 01 65 C5 D9 5D EE51

A680 45 2D 00 26 08 CD 1E 20 : AB
A688 CD E2 1F 4C 4F 4F 50 20 : 28
A690 2D 00 26 0D CD 1E 20 CD : 38
A698 E2 1F 52 45 53 54 20 2D : 8C
A6A0 00 3E 48 21 27 09 CD 1E : C2
A6A8 20 CD F4 1F 26 04 CD 1E : 15
A6B0 20 CD F4 1F 21 25 09 CD : 1C
A6B8 1E 20 3A DB AC CD C1 1F : AC
A6C0 26 0E CD 1E 20 3A DC AC : 01
A6C8 CD C1 1F 21 23 04 CD 1E : E0
A6D0 20 2A D9 AC CD BE 1F C9 : 42
A6D8 21 87 A0 E5 21 00 00 22 : 70
A6E0 D9 AC 3E 01 32 DB AC 3E : BB
A6E8 02 32 DC AC 21 D4 AB 22 : 7E
A6F0 DF AC 3E 00 32 E2 AC 3E : C7
A6F8 0F 32 E1 AC 3E 05 32 E3 : 26
----------------------------------
SUM: 7C 62 9F 27 85 1F 0F 98 27CC

A700 AC 3E 0A 32 E4 AC C3 5B : D4
A708 A6 21 87 A0 E5 21 00 00 : F4
A710 22 D9 AC 3E 01 32 DB AC : 9F
A718 7F 32 DC AC 21 D4 AB 22 : FB
A720 DF AC 3E 00 32 E2 AC 3E : C7
A728 0F 32 E1 AC 3E 03 32 E3 : 24
A730 AC 3E 02 32 E4 AC C3 5B : CC
A738 A6 21 0B 15 CD 1E 20 CD : BF
A740 E2 1F 20 20 20 20 20 20 : C1
A748 20 20 20 20 20 20 20 20 : 00
A750 20 20 20 20 00 CD 1E 20 : 8B
A758 CD E2 1F 53 50 45 45 44 : 3F
A760 20 53 45 4C 45 43 54 20 : 00
A768 5B 41 2D 5A 5D 00 CD CA : 17
A770 1F FE 41 DA 6B A0 D6 40 : 59
A778 32 D6 AC CD C1 1F C3 6B : 8F
----------------------------------
SUM: EE 50 23 AF 6A D6 67 AB 33F3

A780 A0 E5 21 22 12 CD 1E 20 : E5
A788 CD E2 1F 50 41 55 53 45 : 4C
A790 21 00 CD D0 1F FE 20 20 : 1B
A798 F9 CD 1E 20 CD E2 1F 20 : F2
A7A0 20 20 20 20 20 00 E1 C9 : 4A
A7A8 DE A7 E9 A7 F4 A7 FF A7 : 56
A7B0 0A A8 15 A8 20 A8 2B A8 : 0A
A7B8 36 A8 41 A8 4C A8 57 A8 : BA
A7C0 62 A8 6D A8 78 A8 83 A8 : 6A
A7C8 8E A8 99 A8 A4 A8 AF A8 : 1A
A7D0 BA A8 C5 A8 D0 A8 DB A8 : CA
A7D8 E6 A8 F1 A8 FC A8 20 20 : 0B
A7E0 20 1F 1D 1D 1D 20 20 20 : F6
A7E8 00 3C 4F 3E 1F 1D 1D 1D : 3F
A7F0 20 56 20 00 71 4F 70 1F : E5
A7F8 1D 1D 1D 6C 20 6C 00 6F : BE
----------------------------------
SUM: B2 19 EF E0 74 91 EC 48 093E

A800 48 6F 1F 1D 1D 1D 20 55 : A2
A808 20 00 4D 76 4D 1F 1D 1D : 89
A810 1D 59 20 59 00 49 4F 49 : D0
A818 1F 1D 1D 1D 49 20 49 00 : 28
A820 20 55 20 1F 1D 1D 1D 3E : 49
A828 55 3C 00 2A 2A 2A 1F 1D : 4B
A830 1D 1D 2A 2A 2A 00 58 58 : 68
A838 58 1F 1D 1D 1D 58 58 58 : D6
A840 00 2B 2B 2B 1F 1D 1D 1D : F7
A848 2B 2B 2B 00 20 20 20 1F : 00
A850 1D 1D 1D 20 20 20 00 71 : 28
A858 48 70 1F 1D 1D 1D 20 56 : A4
A860 20 00 4D 56 4D 1F 1D 1D : 69
A868 1D 56 20 56 00 20 4F 6F : C7
A870 1F 1D 1D 1D 20 20 20 00 : D6
A878 20 4F 20 1F 1D 1D 1D 20 : 25
----------------------------------
SUM: 9A 57 4C E9 47 3A C7 75 C45D

A880 20 6F 00 20 4F 20 1F 1D : 5A
A888 1D 1D 20 6F 20 00 20 20 : 29
A890 20 1F 1D 1D 1D 6F 4F 20 : 74
A898 00 6F 20 20 1F 1D 1D 1D : 25
A8A0 20 4F 20 00 20 6F 20 1F : 5D
A8A8 1D 1D 1D 20 4F 20 00 20 : 06
A8B0 20 6F 1F 1D 1D 1D 20 4F : 74
A8B8 20 00 20 55 20 1F 1D 1D : 0E
A8C0 1D 2D 55 2D 00 49 55 49 : B3
A8C8 1F 1D 1D 1D 20 55 20 00 : 0B
A8D0 6C 20 6C 1F 1D 1D 1D 64 : D2
A8D8 4F 62 00 20 6F 20 1F 1D : 9C
A8E0 1D 1D 20 20 20 00 20 4F : 09
A8E8 20 1F 1D 1D 1D 2E 20 2E : 12
A8F0 00 2A 4F 2A 1F 1D 1D 1D : 19
A8F8 20 20 20 00 76 20 76 1F : 8B
----------------------------------
SUM: 2E 47 63 4E D5 BD 8C A8 C1F8

A900 1D 1D 1D 20 4F 20 00 64 : 4A
A908 A4 57 A9 01 64 A4 39 AA : 90
A910 02 64 A4 9C A9 03 64 A4 : 5A
A918 A0 A9 04 64 A4 AE A9 05 : B1
A920 FE A3 00 00 06 64 A4 F4 : A3
A928 A9 01 64 A4 15 AA 03 64 : D8
A930 A4 28 AA 04 64 A4 75 A9 : A0
A938 02 22 A4 00 00 05 64 A4 : D5
A940 A4 AA 0D 48 A4 D7 AA 0B : D3
A948 64 A4 43 AB 0C 64 A4 6C : 76
A950 AB 14 52 A4 B3 AB 17 02 : 2C
A958 02 03 03 03 02 02 02 01 : 12
A960 01 01 04 0B 04 04 07 07 : 27
A968 07 08 08 08 09 09 09 06 : 40
A970 06 06 0A 57 A9 02 02 02 : 1C
A978 02 02 05 05 05 02 02 02 : 19
----------------------------------
SUM: 75 E5 E0 D2 9F 25 41 E7 EA32

A980 0B 03 06 06 06 03 02 0B : 30
A988 01 04 04 04 01 02 03 06 : 19
A990 06 06 03 02 01 04 04 04 : 1E
A998 01 0A 7D A9 02 0A 9C A9 : 82
A9A0 05 0C A0 A9 05 0D 0C A0 : 18
A9A8 A9 02 0B 0A A0 A9 02 02 : 0D
A9B0 02 02 0C AE A9 05 0C C0 : 38
A9B8 A9 0B 06 06 06 0A B9 A9 : 32
A9C0 0B 04 04 04 0A C0 A9 0E : 98
A9C8 14 04 04 0E 06 04 04 0C : 44
A9D0 C7 A9 0B 04 04 0C C7 A9 : FF
A9D8 01 02 02 02 0A D9 A9 0E : A1
A9E0 14 06 06 0E 06 06 06 0C : 4C
A9E8 DF A9 0B 06 06 0C DF A9 : 33
A9F0 03 0A D9 A9 02 02 02 02 : 97
A9F8 03 06 09 08 07 0B 04 01 : 31
----------------------------------
SUM: 4C A4 4F F9 91 A0 80 52 1911

AA00 0D 0D 0D 0C F4 A9 02 0C : DE
AA08 F4 A9 02 0C F4 A9 05 05 : 52
AA10 05 05 0A F4 A9 02 02 02 : B7
AA18 02 02 02 02 02 0C 1A AA : DA
AA20 0B 08 05 05 05 0A 15 AA : EB
AA28 0D 0D 0D 0D 0B 0D 0D 0D : 66
AA30 0D 08 08 0B 08 08 0A 28 : 6A
AA38 AA 02 02 02 02 02 02 02 : B8
AA40 0C 3F AA 05 05 05 0E 16 : 28
AA48 0B 05 0C 53 AA 09 09 09 : 34
AA50 0A 4D AA 07 07 07 0A 53 : 73
AA58 AA 02 02 02 02 02 02 02 : B8
AA60 02 01 01 01 01 01 01 01 : 09
AA68 05 05 05 05 0B 07 07 07 : 34
AA70 07 07 07 07 08 08 08 08 : 3C
AA78 08 08 08 08 08 0A 7C AA : 58
----------------------------------
SUM: B8 84 AE A3 81 B2 00 CC 504F

AA80 02 02 02 02 02 02 02 02 : 10
AA88 03 03 03 03 03 03 03 05 : 1A
AA90 05 05 05 0B 09 09 09 09 : 3E
AA98 09 08 08 08 08 08 08 08 : 41
AAA0 08 0A 7C AA 0E 0D 02 05 : 5A
AAA8 0E 0E 02 05 0E 0F 02 05 : 47
AAB0 0E 10 0C C6 AA 01 05 0E : AE
AAB8 11 02 05 0E 12 02 05 0E : 4D
AAC0 13 02 05 0A A4 AA 03 05 : 7A
AAC8 0E 11 02 05 0E 12 02 05 : 4D
AAD0 0E 13 02 05 0A A4 AA 02 : 82
AAD8 02 05 05 0B 05 06 06 06 : 2E
AAE0 06 06 03 03 0B 03 03 03 : 26
AAE8 02 02 02 02 02 01 01 01 : 0D
AAF0 01 01 04 04 04 04 0B 04 : 21
AAF8 04 04 04 04 04 07 07 07 : 29
----------------------------------
SUM: 86 74 BC C7 C4 AA EF 5F 5A4F

AB00 07 07 08 08 0B 08 08 09 : 42
AB08 06 06 06 06 02 02 02 02 : 20
AB10 05 08 08 0B 08 08 08 08 : 40
AB18 08 08 08 06 06 05 04 01 : 2E
AB20 01 01 01 01 02 02 02 02 : 0C
AB28 02 06 06 0B 06 06 06 05 : 30
AB30 05 06 06 06 06 07 07 08 : 33
AB38 08 0B 08 08 08 08 09 09 : 45
AB40 0A DB AA 05 05 0D 0D 0D : C0
AB48 0D 05 05 05 06 06 06 04 : 32
AB50 04 04 04 04 04 06 06 06 : 26
AB58 06 06 04 04 04 04 0B 06 : 2D
AB60 06 06 06 0B 04 04 08 08 : 35
AB68 0B 0A 43 AB 0D 05 05 0D : 27
AB70 0E 15 05 05 0D 05 05 02 : 46
AB78 02 02 02 02 02 02 02 02 : 10
----------------------------------
SUM: 6C 46 3A 08 64 5B 66 62 5F3E

AB80 02 02 02 02 02 0D 0D 0E : 32
AB88 14 05 05 08 05 08 05 08 : 40
AB90 0B 05 08 05 08 05 08 05 : 37
AB98 08 05 08 0B 05 08 05 08 : 3A
ABA0 05 08 05 08 05 08 05 08 : 34
ABA8 05 08 05 08 05 08 05 08 : 34
ABB0 0A 6C AB 05 05 0E 18 05 : 56
ABB8 05 0E 19 05 05 0E 1A 05 : 63
ABC0 05 0C CC AB 03 03 03 03 : 94
ABC8 0B 0A C4 AB 01 01 01 01 : 88
ABD0 0B 0A CC AB 0B 0A 00 32 : D3
ABD8 02 05 00 32 02 05 00 64 : A4
ABE0 01 0A 00 64 0B 04 00 14 : 92
ABE8 03 0A 00 96 01 04 03 02 : AD
ABF0 00 32 0D 01 00 64 06 06 : B0
ABF8 00 32 01 0A 00 32 01 04 : 74
----------------------------------
SUM: 63 38 4F 6C 45 FF 69 F7 04E0

AC00 07 06 00 96 06 05 00 64 : 12
AC08 03 11 00 64 04 07 00 96 : 19
AC10 03 04 08 04 03 0A 08 02 : 2A
AC18 03 05 00 32 0D 01 00 96 : DE
AC20 02 0E 00 32 04 05 00 1E : 69
AC28 05 05 00 1E 06 04 00 64 : 96
AC30 0E 04 00 96 0C 0A 00 32 : F0
AC38 0B 04 00 96 03 14 00 32 : EE
AC40 0E 08 00 96 10 04 00 4B : 0B
AC48 01 05 00 1E 07 0A 00 4B : 80
AC50 03 0A 00 1E 05 0A 00 96 : D0
AC58 02 14 0A 0A 00 4B 0A 0A : 89
AC60 00 4B 09 04 00 4B 0C 1E : CD
AC68 00 32 0F 02 00 96 0E 0A : F1
AC70 00 32 0F 0A 00 32 10 0F : 9C
AC78 00 64 0C 05 00 14 10 04 : 9D
----------------------------------
SUM: 44 79 45 9D 4F C8 4C E9 E320

AC80 00 32 10 02 00 1E 10 02 : 74
AC88 00 32 0F 04 00 32 09 05 : 85
AC90 00 14 0A 0A 00 32 0E 05 : 6D
AC98 00 96 03 1E 08 05 00 32 : F6
ACA0 02 14 08 02 02 05 00 4B : 72
ACA8 07 0A 05 05 06 04 01 04 : 2A
ACB0 0F 01 00 32 09 02 00 96 : E3
ACB8 01 0A 02 0A 03 0A 04 0A : 32
ACC0 0C 28 00 78 0E 06 00 96 : 56
ACC8 0F 0A 00 32 10 04 00 28 : 87
ACD0 05 0A 00 96 FF FF 19 01 : BD
ACD8 00 0E 16 2C AC 01 00 03 : 00
ACE0 04 78 00 00 00 0F C6 A9 : FA
ACE8 00 00 00 00 0F 16 D6 A9 : A4
ACF0 00 00 00 00 0A 16 D6 A9 : 9F
ACF8 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: 3D F9 51 DD FE E1 B7 EA F508

AD00 00 00 00 00 00 00 00 00 : 00
AD08 00 00 00 00 00 00 00 00 : 00
AD10 00 00 00 00 00 00 00 00 : 00
AD18 00 00 00 00 00 00 00 00 : 00
AD20 00 00 00 00 00 00 00 00 : 00
AD28 00 00 00 00 00 00 00 00 : 00
AD30 00 00 00 00 00 00 00 00 : 00
AD38 00 00 00 00 00 00 00 00 : 00
AD40 00 00 00 00 00 00 00 00 : 00
AD48 00 00 00 00 00 00 00 00 : 00
AD50 00 00 00 00 00 00 00 00 : 00
AD58 00 00 00 00 00 00 00 00 : 00
AD60 00 00 00 0C 06 00 0D 10 : 2F
AD68 00 08 00 00 04 17 00 0D : 30
AD70 17 00 0E 17 00 1B 17 00 : 6E
AD78 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: 17 08 0E 23 0A 32 24 1D 62B6

AD80 00 00 00 00 00 00 00 00 : 00
AD88 00 00 00 00 00 00 00 00 : 00
AD90 00 00 00 00 00 00 00 00 : 00
AD98 00 00 00 00 00 00 00 00 : 00
ADA0 00 00 00 00 00 00 00 00 : 00
ADA8 00 00 00 00 00 00 00 00 : 00
ADB0 00 00 00 00 00 00 00 00 : 00
ADB8 00 00 00 00 00 00 00 00 : 00
ADC0 00 00 00 00 00 00 00 00 : 00
ADC8 00 00 00 0C 15 13 05 10 : 49
ADD0 16 0E 0B 15 09 06 04 00 : 57
ADD8 05 1D 15 1B 0C 18 00 11 : 87
ADE0 16 0D 01 1C 02 16 16 1F : 8D
ADE8 03 08 0C 02 01 01 05 09 : 29
ADF0 0C 0B 01 07 0A 04 0A 0F : 46
ADF8 16 12 15 0A 07 0B 0E 13 : 7A
----------------------------------
SUM: 56 5D 43 6B 3E 57 3C 6B 86E4

AE00 11 0A 72 48 00 09 E0 22 : E0
AE08 00 00 00 01 00 00 23    : 24
----------------------------------
SUM: 11 0A 72 49 00 09 03 22 A155
```

▶夏のボーナスの一部を中国ファンドでふくらまし，ついにX68030を買ってしまった。
中嶋　祥史(27)神奈川県

命令のクロックは?(PART3)

●シフト，ローテイト命令

・シフトレフトアリスメチック，算術左シフト命令(シフトライトアリスメチックSRA命令，シフトライトロジカルSRL命令も同様)

SLA A〜L

以上，8クロック

SLA (HL)

以上，15クロック

SLA (IX+d)
SLA (IY+d)

以上，23クロック

・アキュムレータローテイト命令

RLA
RLCA
RRA
RRCA

以上，4クロック

・ローテーレフト命令（ローテーレフトサーキュラRLC命令，ローテーライトRR命令ローテーライトサーキュラRRC命令)

RL A〜L

以上，8クロック

RL (HL)

以上，15クロック

RLA (IX+d)
RLA (IY+d)

以上，23クロック

・ローテーレフト（ライト）デジット命令

RLD
RRD

以上，18クロック

●比較インクリメント（デクリメント）命令

CPD
CPI

以上，16クロック

CPDR
CPIR

以上，1バイトにつき21クロック，最終のみ16クロック

●I/O入出力命令

・間接アドレッシング入力命令

IN A〜L,(C)

以上，4クロック

・直接アドレッシング入力命令

IN A,(n)

以上，11クロック

・ブロック入力命令

IND
INI

以上，16クロック

INDR
INIR

以上，1バイトにつき21クロック，最終のみ16クロック

・間接アドレッシング出力命令

OUT (C),A〜L

以上，12クロック

・直接アドレッシング出力命令

OUT (n),A

以上，11クロック

・ブロック入力命令

OUTD
OUTI

以上，16クロック

OTDR
OTIR

以上，1バイトにつき21クロック，最終のみ16クロック

●ビット操作命令

・ビットテスト命令

BIT 0〜7,A〜L

以上，8クロック

BIT 0〜7,(HL)

以上，12クロック

BIT 0〜7,(IX+d)
BIT 0〜7,(IY+d)

以上，20クロック

・ビットセット命令

SET 0〜7,A〜L

以上，8クロック

SET 0〜7,(HL)

以上，15クロック

SET 0〜7,(IX+d)
SET 0〜7,(IY+d)

以上，23クロック

・ビットリセット命令

RES 0〜7,A〜L

以上，8クロック

RES 0〜7,(HL)

以上，15クロック

RES 0〜7,(IX+d)
RES 0〜7,(IY+d)

以上，23クロック

●交換命令

EX (SP),HL

以上，19クロック

EX (SP),IX
EX (SP),IY

以上，23クロック

EX AF,AF'
EX DE,HL
EXX

以上，4クロック

●割り込み関係

・割り込み禁止

DI

以上，4クロック

・割り込み許可

EI

以上，4クロック

・割り込みモード設定

IM 0〜2

以上，8クロック

・割り込み処理からのリターン

RETI
RETN

以上，14クロック

●リスタート命令

RST nnH
(nn=00,08,10,18,20,28,30,38)

以上，11クロック

●その他

SCF
HALT
NOP

以上，4クロック

▶ 全機種共通システムインデックス ◀

＊以下のアプリケーションは，基本システムであるS-OS“MACE”またはS-OS“SWORD”がないと動作しませんのでご注意ください。

▶「スターラスター」のオリジナルからアレンジへ替えたときの視界の広がりは，「ガンダム」のコックピットの視界から「Zガンダム」のそれに替わったときと同じ感じがすると思いませんか？
伊藤　充(20)神奈川県

第22部 magiFORTH TRACER
第23部 ディスクダンプ＆エディタ
第24部 "SWORD" 2000 QD
連載 対話で学ぶmagiFORTH
特別付録 PC-8801版S-OS "SWORD"
■86年7月号
第25部 FM音源ミュージックシステム
付録 FM音源ボードの製作
連載 計算力アップのmagiFORTH
特別付録 SMC-777版S-OS "SWORD"
■86年8月号
第26部 対局五目並べ
第27部 MZ-2500版S-OS "SWORD"
■86年9月号
第28部 FuzzyBASIC発表
連載 明日に向かってmagiFORTH
■86年10月号
第29部 ちょっと便利な拡張プログラム
第30部 ディスクモニタDREAM
第31部 FuzzyBASIC料理法<1>
■86年11月号
第32部 パズルゲームHOTTAN
第33部 MAZE in MAZE
連載 FuzzyBASIC料理法<2>
■86年12月号
第34部 CASL & COMET
連載 FuzzyBASIC料理法<3>

1987

■87年1月号
第35部 マシン語入力ツールMACINTO-C
連載 FuzzyBASIC料理法<4>
■87年2月号
第36部 アドベンチャーゲームMARMALADE
第37部 テキアベ作成ツールCONTEX
■87年3月号
第38部 魔法使いはアニメがお好き
第39部 アニメーションツールMAGE
付録 "SWORD" 再掲載とMAGICの標準化
■87年4月号
第40部 INVADER GAME
第41部 TANGERINE
■87年5月号
第42部 S-OS "SWORD" 変身セット
第43部 MZ-700用 "SWORD" をQD対応に
■87年6月号
インタラプト コンパイラ物語
第44部 FuzzyBASICコンパイラ
第45部 エディタアセンブラZEDA-3
■87年7月号
第46部 STORY MASTER
■87年8月号
第47部 パズルゲーム碁石拾い
第48部 漢字出力パッケージJACKWRITE
特別付録 FM-7/77版S-OS "SWORD"
■87年9月号
第49部 リロケータブル逆アセンブラInside-R
特別付録 PC-8001/8801版S-OS "SWORD"
■87年10月号
第50部 tiny CORE WARS
第51部 FuzzyBASICコンパイラの拡張
第52部 XIturbo版S-OS "SWORD"
■87年11月号
序論 神話のなかのマイクロコンピュータ
付録 S-OSの仲間たち
第53部 もうひとつのFuzzyBASIC入門
第54部 ファイルアロケータ＆ローダ
インタラプト S-OSこちら集中治療室
第55部 BACK GAMMON
■87年12月号
第56部 タートルグラフィックパッケージTURTLE
第57部 XIturbo版 "SWORD" アフターケア ラインプリントルーチン
特別付録 PASOPIA7版S-OS "SWORD"

1988

■88年1月号
第58部 FuzzyBASICコンパイラ・奥村版
付録 石上版コンパイラ拡張部の修正
■88年2月号
第59部 シューティングゲームELFES
■88年3月号
第60部 構造型コンパイラ言語SLANG
■88年4月号
第61部 デバッギングツールTRADE
第62部 シミュレーションウォーゲームWALRUS
■88年5月号
第63部 シューティングゲームELFESⅡ
第64部 地底最大の作戦
■88年6月号
第65部 構造化言語SLANG入門(1)
第66部 Lisp-85用NAMPAシミュレーション
■88年7月号
第67部 マルチウィンドウドライバMW-1
連載 構造化言語SLANG入門(2)
■88年8月号
第68部 マルチウィンドウエディタWINER
■88年9月号
第69部 超小型エディタTED-750
第70部 アフターケアWINERの拡張
■88年10月号
第71部 SLANG用ファイル入出力ライブラリ
第72部 シューティングゲームMANKAI
■88年11月号
第73部 シューティングゲームELFESⅣ
■88年12月号
第74部 ソースジェネレータSOURCERY

1989

■89年1月号
第75部 パズルゲームLAST ONE
第76部 ブロックゲームFLICK
■89年2月号
第77部 高速エディタアセンブラREDA
特別付録 XI版S-OS "SWORD"<再掲載>
■89年3月号
第78部 Z80用浮動小数点演算パッケージSORBAN
■89年4月号
第79部 SLANG用実数演算ライブラリ
■89年5月号
第80部 ソースジェネレータRING
■89年6月号
第81部 超小型コンパイラTTC
■89年7月号
第82部 TTC用パズルゲームTICBAN
■89年8月号
第83部 CP/M用ファイルコンバータ
■89年9月号
第84部 生物進化シミュレーションBUGS
■89年10月号
第85部 小型インタプリタ言語TTI
■89年11月号
第86部 TTI用パズルゲームPUSH BON!
■89年12月号
第87部 SLANG用リダイレクションライブラリDIO.LIB

1990

■90年1月号
第88部 SLANG用ゲームWORM KUN
特別付録 再掲載SLANGコンパイラ
■90年2月号
第89部 超小型コンパイラTTC++
■90年3月号
第90部 超多機能アセンブラOHM-Z80
■90年4月号
第91部 ファジィコンピュータシミュレーションI-MY
■90年5月号
第92部 インタプリタ言語STACK
■90年6月号
第93部 リロケータブルフォーマットの取り決め
第94部 STACK用ゲームSQUASH!
第95部 X68000対応S-OS "SWORD"
特別付録 PC-286対応S-OS "SWORD"
■90年7月号
第96部 リロケータブルアセンブラWZD
■90年8月号
第97部 リンカWLK
■90年9月号
第98部 BILLIARDS
■90年10月号
第99部 ライブラリアンWLB
■90年11月号
第100部 タブコード対応エディタEDC-T
■90年12月号
第101部 STACKコンパイラ

1991

■91年1月号
第102部 ブロックアクションゲームCOLUMNS
■91年2月号
第103部 ダイスゲームKISMET
■91年3月号
第104部 アクションゲームMUD BALLIN'
■91年4月号
第105部 SLANG用カードゲームDOBON
■91年5月号
第106部 実数型コンパイラ言語REAL
■91年6月号
第107部 Small-C処理系の移植
■91年7月号
第108部 REALソースリスト編
■91年8月号
第109部 Small-Cライブラリの移植
■91年9月号
第110部 SLANG用NEWファイル出力ライブラリ
■91年10月号
第111部 Small-C活用講座（初級編）
■91年11月号
第112部 Small-C活用講座（応用編）
第113部 MORTAL
■91年12月号
第114部 Small-C SLANGコンパチ関数

1992

■92年1月号
第115部 LINER
■92年2月号
第116部 シミュレーションゲームPOLANYI
■92年3月号
第117部 カードゲームKLONDIKE
■92年4月号
第118部 オプティマイザO80実践Small-C講座(1)
■92年5月号
第119部 COMMAND.OBJ実践Small-C講座(2)
■92年6月号
第120部 COMMAND.OBJ2実践Small-C講座(3)
■92年7月号
第121部 関数リファレンス実践Small-C講座(4)
■92年8月号
第122部 ワイルドカード実践Small-C講座(5)
第123部 グラフィックライブラリ GRAPH.LIB
■92年9月号
第124部 O-EDIT&MODCNV
■92年10月号
第125部 SLENDER HUL実践Small-C講座(6)
■92年11月号
第126部 EDIT実践Small-C講座(7)
■92年12月号
第127部 MAKE実践Small-C講座(8)

1993

■93年1月号
第128部 EDC-Tの拡張
■93年2月号
第129部 BLACK JACK
■93年3月号
第130部 シューティングゲームコアシステム作成法(1)
■93年4月号
第131部 シューティングゲームコアシステム作成法(2)
■93年5月号
第132部 シューティングゲームコアシステム作成法(3)
■93年6月号
第133部 REVERSI
■93年7月号
特別付録 MSX用S-OS "SWORD"
■93年8月号
第134部 MACINTO-C再掲載
■93年9月号
第135部 7並べ
特別付録 SLANG再々掲載
■93年10月号
第136部 シューティングゲームコアシステム作成法(4)
■93年11月号
第137部 S-OSで学ぶZ80マシン語講座(1)
■93年12月号
第138部 エディタアセンブラREDA再掲載

[第7回]

蓄積感を持たない人々

Ogikubo Kei
荻窪 圭

最近，見るテレビはといえば，Jリーグの中継か，それに類するスポーツニュースしかないなぁと思う今日この頃である。Jリーグといえば，アレね，アレ。菊池審判。グランパス－ヴェルディ戦。得点に直接関係するミスジャッジが2つもあって，どちらもヴェルディに有利に働いた，って点で，いろいろといわれている。私もテレビ見ながら，「ナンナンダ，アレハ」って憤っていたわけだ。審判のレベルがどうとか，っていうけど，プロの試合なんだからさ，勝ち負けが直接収入その他に響いてくるわけで，許せないミスジャッジに属する。ファール（となるべきプレイ）があり，審判はそれに気づかず，その直後のプレイで得点が入った。まあ，見逃しは仕方がない。怒れば済むことである。問題は，点が入ったあと，先のファールを認めてイエローカードを出し（レッドカードを出すべきだったという意見もあるが）たのに，ファールのあとの得点はそのまま認めた，ってことにある。これ，論理的な矛盾ではないだろうか。ファールのあった時点までプレイを戻して再開するのが筋ではないだろうか。普通，ファール（となるべきプレイ）があり，それをちょっと流したあとで，ファールをとったとしたら，ファールの時点までボールを戻して再開するでしょ。そういうシーンは何度も見たことがある。今回のは特に，得点が絡んでいるわけで，審判はより慎重でなければならない。審判がプロだからとかプロでないからとか，そういう問題ではないわけだ。

で，腹が立ったので，メガドラ2の電源を入れ（いまごろメガドラ2なんて買っているのだ，あたしゃ），「プロストライカー2」を立ち上げる。これ，メガドラ用のJリーグ公認ゲームね。そもそもテレビをあまり見なくなった理由のひとつがこれだ。何しろ，テレビ番組を見ているより「ぷよぷよ」やっている時間のほうが長いのだから（とほほ）。

で，プロストライカー2だが，これがまたバカでさ。バカなの。味方選手の動きがどう考えてもバカ。どうして敵が左サイドを上がってきているのに，左サイドバックがタッチラインの外にいるわけ？　どうしてペナルティエリアに敵が入ってきてピンチのときにディフェンダーがゴールラインの外に出ているわけ？　これじゃあディフェンダーの意味がないじゃん。あまりにバカすぎる。しかも，パスもバカでさ，誰もいないところに蹴ったりしてるの。パス出そうとしても出す相手がいないしさ。

考えてみたら，これ，モノホンのグランパスとあまり変わりないじゃないか，って思ったりするのだが（ヴェルディでやると，なぜかそれなりにパスがつながるのだ。くそぅ），それにしても，選手の動きにインテリジェントさがなさすぎる。

たとえば，4-4-2システムでディフェンダーを4人置く。これがどのチームも基本なのだが，おかげで，中央突破が楽なのだ。両サイドから攻めようとすると，サイドバックが常に守っているからやりにくいけど，2～3人でショートパス回しながら中央突破すると，ディフェンダーふたりの間をうまく抜けてゴール前に行けるのだ。守備のときは逆ね。中央からこられると，どっちのディフェンダーを先に反応させればいいかわからなくなって，あっさり突破されてしまう。バカ。ボールが近くにあっても，自分のポジションに戻ることを優先するヤツもいるし。どうも賢くない。GKが賢いから，点がまり入らないってだけなのだ。

PC用の（スーファミ用も出ていたけど）「World Cup USA'94」だと，見た目は同じようなものなのに，さすが，ちゃんと動いてくれる。パスも，パスを出す相手を選んで蹴る方式で，方向だけを定めてエイヤ，って蹴るプロストライカー2とは違う。よくできている。

面白いのが，フォーメーションを登録する機能。4-4-2でも，味方がボールをキープしていてフィールド内のどのエリアにボールがあれば選手はどう動いて，逆に敵がボールをキープしているときは味方はこのポジションを取る，ってことがちゃんと設定できる。コーナーキック時の選手の配置も設定できる。戦術を考える楽しみがあるわけだ。まあ，さほど賢い動きをしてくれるわけではないが，プロストライカー2よりはマシだ。

見た目（つまり，画面構成）は似ているし，選手の動きのリアルさや凝り方は圧倒的にプロストライカー2のほうが上なのだが，戦術的にもインテリジェントさでもリアルさでも負けるのだ。見かけだけはリアルでカッコいいなんて，非常に日本的である。エンディング（つまり，優勝したとき）の仰々しさでもプロストライカー2が上だが，ゲーム内容があれでは虚しい限りだ。

別に，プロストライカー2の悪口がいいたいのではなくて（だったら，2シーズンもフルに遊んだりしない），えっと，次をどうぞ。

処理と蓄積と共有

それよりも，PCのサッカーゲームとメガドラのサッカーゲームで決定的に違うと思ったのは，「蓄積感」だ。自

分の行為が「蓄積されている」という感覚。確かに，プロストライカー2でもセーブ機能はあって，ちゃんと結果は保存される。でも，違う。プロストライカー2では使ったフォーメーションとか交代選手が保存されない。このくらい，何十バイトかあれば記録できるのだからなんとかなりそうなものだがなぁ。毎回フォーメーションを変えて，リザーブとレギュラーを変えて，ってやるのは異様にかったるい。チームによってプレイしやすいフォーメーションもあるのだし。

それだけではない。たとえば，得点ランキングに出てくる選手や，どのチームが強いか弱いかといった順位の変動にリアルさがないのだ。だから，自分のやったゲームの勝ち負けしか蓄積されないという感じがつきまとう(実際にそうなのだが，そうは感じさせないのが技というものだろうに)。こういうゲーム(この場合は，リーグ戦)って，「自分が手を加えてひとつの歴史を作っていく」って面白さもあるわけだから，「蓄積感」の有無は致命的な問題ではないかと思う。PCのサッカーゲームでは，完璧とはいえないにしろ，ちゃんと「蓄積感」がある。自分が勝ち進んだ歴史が残っている，という感覚だ。World Cup USA '94では自分の考えたフォーメーションや戦術が，そのまま保存されている。それだけでも，自分の行為が蓄積されていく面白さがある。

これは，大量のハードディスクを使えるパソコンゲームと，セーブできるデータ量に限界のあるゲーム専用機の違いといっていい。

昔，NECの富田本部長（だったかな）に取材にいったとき，パソコンの発展には3つの段階がある，という話を聞いた。ずいぶん前の取材だったからよく覚えてないけど，確か，「処理」—「蓄積」—「共有」の3つだったと思う。

最初のパソコンというのは，データを入れて処理して吐き出すだけの存在だった，ってこと。データが必要なとき，その都度入れる。「処理」こそがコンピュータだ，というわけ。ゲーム機や多くのワープロ専用機はこの段階だ。ゲーム機でいえば，虚しさは残るけれども，その場が楽しければいい，みたいな感じ。イっちゃえばそれで幸せ，みたいな感じ。アーケードゲームなんかその最たるものね。

これがハードディスクの普及により，「蓄積」の時代となった。データは常にファイルという形で蓄積され，それが処理される。新しく入力したデータは処理されて出力されると同時に蓄積される。

現在はこの段階ね。ユーザーは常に自分の作ったデータやどこかから持ってきたデータが蓄積されるのを意識している。データベース的発想だ。データは比較され，使い回され，生きていく。まあ，ほとんどは死蔵されたりするのだが，蓄積されているという安心感は大きい。

その次は共有の時代。自分で蓄積するだけではなく，点在するデータを自在に探し，自在に入手し，自在に提供するようになる。蓄積されたものをネタにコミュニケーションする時代といっていいかもしれない。

蓄積感覚なき共有の時代とは

いま，パソコンを持たない，あるいはパソコンで何らかの蓄積をしていくという「デジタルデータ蓄積感覚」が身についていない多くの人たちがいる。これは間違いなく，多くいる。

現在，情報スーパーハイウェイに触発されてネットワーク論議が盛んだが，もしこれが実現されると，「蓄積感覚」を持たない人がいきなり「共有」の世界へと飛び込むわけだ。確かに，共有すれば手元に蓄積する必要は減るだろうが，蓄積感覚のないネットワークはただ単に「情報の自動販売機」と化するだけで，ロクなものではない。それじゃ，何のためのネットワークだ。結局，中心があって分配されるという，ワイヤードなデジタルメディアができあがるだけではないか。これでいいのだろうか。

ということを，メガドラで遊びながら考えてしまったわけだ。別に「ぷよぷよ」に蓄積感など求めないが，やはり，蓄積されていく感覚を楽しむゲームもあれば，その場その場の瞬間を楽しむ刹那的なゲームもある。Jリーグものやワールドカップものは，両方を同時に満たさなければならない世界だ。

仮想的な世界に，自らが手を加えることによって歴史が蓄積される，という感覚。これは何ものにも代え難い面白さではないかと思うのだ。ね，思うでしょ。

が，しかし，である。うーむ。考える，考える，考える。いや，考えてはいけないのかもしれない。感じる，感じる，感じる。

パソコンというのは，もともとコンピュータと人間の1対1の関係をベースに発達してきた。いまでも多くのパソコンユーザーは1対1の関係をベースにしてパソコンと接している。「蓄積」の面白さはそこにある。パソコンユーザーは自分のハードディスクにデータを蓄積することによって，そのファイルを自分の所有物と感じる，とよくいわれる。仮に，他人が作ったファイルであっても，ネットワークから拾ってきたファイルであっても，自分のハードディスクに収まった瞬間，それは所有物なのだ。そういう感覚である。普通の人が，たとえばテレビ番組を録画したテープをどれだけ大量に持っていても，その番組が自分の所有物だという感覚にはならない。しかし，パソコンユーザーにとっては，ファイルはどんなものでも所有物なのだ。それは，その情報がハードディスクにコピーされた途端，元情報から切り離された，出自から切り離された，独立した存在となるからにほかならず，他から独立したものに対して，ユーザーは加工や編集などその情報をいじる自由を獲得するからではないだろうか。

このあたりはパソコンユーザーの特質として特筆されるべきだろう。

その場の楽しさ，トリップ感覚を取るか，クールに蓄積していく歴史の面白さを取るか。これはどちらがいいという問題ではなく，どちらも重要な楽しみだ。

しかし，ここに「共有」という問題もある。

いままでは，「蓄積」のあとにくる「共有」が正しいと思われていた。パソコン通信もそれを売りにしようとしていた。つまり，知識や情報の共有だ。仕事を進めるのに必要な資料や情報，ノウハウを効率よく入手するための「共有」である。人々が蓄積しあったデータを共有することによって，デジタルデータをもっと効率よく使おう，ってわけだ。その考えは，常にパソコンを意識したものとなっている。自分とパソコンの１対１の関係があり，そのパソコンの向こうにネットワークがある，という感覚だ。そこに流れるのは知識である。なんというか，非常にレベルの高い世界である。レベルの高い人たちが密度の濃いレベルの高い情報を共有するわけだ。それは素晴らしいことである。

逆に，「蓄積」をすっとばした「共有」が何を生むか。蓄積感覚がないと，ネットワークが「情報の自動販売機」になってしまう。それは，どう転んでもそうなると思う。そういった意味では，現に，インターネットに手を出し始めているいろいろな企業がしているのがmosaicで利用できるwwwサーバーの設置だということなどが，その典型だ。wwwはハイパーテキストな構造やテキスト以外のデータもサポートしており，マウスでもって視覚的にコントロールできる。マスコミがインターネットを紹介するときは，たいていmosaicを例に挙げている。まだmosaicの利用者（インターネットの利用者という意味ではなく，インターネットのwwwを利用できる人，って意味だが）は少ないけれども，これから爆発的に増えていくだろうという読みなのだろう。

そしてもうひとつ，「情報の共有」よりも「コミュニケーション」を主体にしたネットワークが作られていくだろう。これは私の「読み」だ。情報を蓄積し，共有し，知識を利用しあうよりも，その新しい場を使った，瞬間瞬間のコミュニケーションの面白さを求める，ということ。

情報の蓄積だとか専門的な情報は，誰もが必要としているわけではなくて，多くの人はもっと素人レベルで趣味の話をしたいと願う。どんどん利用者が増えていけば，発信される情報も増えるし，それに従って，情報の質は玉石混淆になり，密度も落ちていく。そういうものだ。

これは，パソコンゲームの持つ「蓄積感」とゲーム専用機の持つ「リアルタイム性」との対比で語られる。

どっちが優勢か。明らかに後者ではなかろうか。

しかも，パソコン通信のソフトはどんどん簡単になり，GUIベースで，マウスでもってアイコンをクリックするだけでサイバースペースを渡り歩くことが可能になってきている。この傾向はどんどん加速する。すると，より多くの人がより気軽にサイバースペースに入り込むことが可能になる。

GUIベースの通信ソフトは，オートパイロットでもってまとめてダウンロードし，オフラインで編集してブラウジングしてレスをつけ，またオートパイロットでアクセスする，というオフライン型の通信には向かない。もっと刹那的な，その場その場のノリでやれる，オンラインですべて処理できてしまうオンライン型の通信に向いている。オンライン型の通信では，じっくり内容を考えて書いてアップするより，その場の思いつきや感覚的な書き込みを促進する。つまり，電話やらポケベルやら伝言ダイヤルやらパーティダイヤルのノリでコミュニケーションできるのだ。

機械を介したコミュニケーションを苦とせず，そればかりか，機械を介したコミュニケーションを楽しんでしまう世代というのは，明らかに存在する。それはゲームボーイとポケベルをカバンに入れた中高生たちの世代であり，下手をしたら携帯電話ですら持ち歩いている。おそらく，PDAが登場し，もし本当に，無線でもって，どこででもサイバースペースにアクセスできるとしたら，パソコン通信における口コミ型のコミュニケーション形態である（NIFTYでいう）ホームパーティが，爆発的に普及するだろう。明らかにそうなると思う。いままでの口コミによる仲間内だけの情報の流布がサイバースペースにも入り込んでくるわけだ。

表の窮屈さと裏の面白さ

要するに，実社会の流れがサイバースペースにも持ち込まれていくということだ。いまはそうでもないけれども，今後，通信人口が増え，誰もがネットワークにアクセスできる時代になると，確実にそうなっていくだろう。

しかし，どんな形でそうなるのかはわからない。たぶん，大手のネットワークでは表と裏ができるだろう。NIFTYでいえば，表というのは公開されたフォーラムで，裏とは非公開のホームパーティだ。何万人もが目にする表の世界の窮屈さから逃れ，仲間内だけの気を使わない言語で書ける裏の世界で楽しむ，といってもいい。

表は窮屈である。最近，特にそれを感じる。誰が読んでいるかわからないから，感情ほとばしることを書くわけにはいかないし，揚げ足を取られないように慎重に言葉を選び，読む人のことを考えなければならない。当たり前といえば当たり前だが，窮屈といえば窮屈である。もっと書きたいように書きたいと思う人も多いはずだ。そういう人は，裏へと潜る。クローズドな世界なら，誰にもとがめられず，好きなことが書ける。

もしかしたら，すでにそうなっているのではないか，という気すらしている。ホームパーティの実体というのは外からは見えないアンダーグラウンドだから。

具体例をあまり挙げられなくて申し訳ないが（そもそも，地下の話なのだから），全国規模の巨大なネットワークの面白さがあり，小さな草の根ネットの面白さがある。巨大な大手ネットワーク内に，ホームパーティのような形で草の根的なムラができていると考えてもいいだろう。

情報の共有よりもコミュニケーションを楽しもうとなると，ある程度仕方ないことかもしれない。いま，ネットで本当に面白いのは，そういうアンダーグラウンドの世界なのかもしれないのだ。

TeX入門講座 〜てふてふらてふ〜

数式の巻

Taki Yasushi 瀧 康史

TeXを使ってなにか印刷してみましたか？
今月はTeXの大きな特長のひとつである数式を扱ってみます
理系でレポートを書く方は見逃さないように

数式を書く

先月号で，TeXへの取っかかりが掴めたと思います。でも，まだまだ自由にTeXでの文整形は難しいといったところでしょうか？　うまくスペースを空けようにも，勝手なスペースやリターンは消されちゃいますし。

TeXをやる醍醐味のひとつに，「勝手に整形をしてくれる」ということがあります。だから，とりあえず最初のうちは，文整形に関しては，すべてTeXシステムにお任せして，慣れてきたら，徐々に自分自身のエゴを含めた勝手なスタイルで印刷をするコツを掴んでいくのが，TeXの楽なマスター方法だと思います。

もっとも，ここではTeXに触れてみるというのが大きな目標なので，スタイルに関してまで扱うかはわかりませんけど。

そんなわけで，今月は数式を書くことに関するお話。理系で数式がたくさん出てくるようなレポートを書く人は，参考にしてください。

前回はなんにもいいませんでしたが，私がここで扱うのは，ことわりがない限り，LaTeXのことであって，TeXそのものに対して説明しているわけではありません。だから今回の数式も，LaTeXよりもっと数式に適したTeXである，Ams-LaTeX*1やLamsTeX*2のことではなくて，LaTeX上での数式の扱いですから，念のため。ちなみにLaTeXとはplain TeXのマクロ集として定義される，使いやすさに重点をおいたシステムです。

＊1，＊2　どちらも，数式に関して強化されたTeXであるAms-TeXと，文整形に強力な，お馴染みLaTeXを組み合わせたものです。ベースになったのがどちらであるか？　という違いしかなかったはずです。

数式モード（印字例1）

LaTeXで数式を使うためには，数式モードに移行しなくてはなりません。

この数式モードには，大きく分けて2種

印字例2（69%に縮小，プリンタはBJC-400Jを使用）

2　演算子などなど

2.1　四則演算子

TeX では、以下のようにして、四則演算を表記します。

$$4+5\times 6-7\div 8=33.125$$

法則性としては、BASICやCなどで数式を表記するのと似てるでしょ？

2.2　分数表記

分数は一般的に、このように表記します。

$$\frac{b}{a}$$

このような感じです。
当然こういうこともできます。

$$\frac{1+\frac{a}{b}}{c}$$

2.3　根号（ルート）

ルートは、このようになります。

$$\sqrt{\frac{a}{b}-c\times\sqrt{a\cdot b\cdot c}}$$

このように、ルートのなかにルートを入れるもできるわけです。
なかに分数などがある場合、LaTeX が自動的に大きさを変えてくれます。

リスト2

```
 1: ¥section{演算子などなど}
 2:
 3: ¥subsection{四則演算子}
 4:
 5: ¥TeX では、以下のようにして、四則演算を表記します。
 6:
 7: ¥[4+5 ¥times 6 - 7 ¥div 8 = 33.125¥]
 8:
 9: 法則性としては、BASICやCなどで数式を表記するのと似てるでしょ？
10:
11: ¥subsection{分数表記}
12:
13: 分数は一般的に、このように表記します。
14:
15: ¥[ ¥frac{b}{a} ¥]
16:
17: このような感じです。
18:
19: 当然こういうこともできます。
20:
21: ¥[ ¥frac{1+¥frac{a}{b}}{c} ¥]
22:
23: ¥subsection{根号(ルート)}
24:
25: ルートは、このようになります。
26:
27: ¥[ ¥sqrt{¥frac{a}{b}-c ¥times ¥sqrt{ a ¥cdot b ¥cdot c}} ¥]
28:
29: このように、ルートのなかにルートを入れるもできるわけです。
30:
31: なかに分数などがある場合、¥LaTeX が自動的に大きさを変えてくれます。
```

類あります。ひとつは，文中に挟むインライン数式モードで，もうひとつは別組みで表す，ディスプレイ数式モードです。

同じ式を，1.1と1.2に表してみました。リスト1と見比べてください。

どうです？　2つのモードの使い分けは読者の皆さんの書くレポートの種類によりけりだと思います。普通，インライン数式モードでは適当な簡単な式を，ディスプレイ数式モードでは重要な式を表示するかと思います。

リストでの違いは，数式モードへの入り方だけです。まず1.1のインライン数式モードは，＄で入って＄で抜けています。そして1.2のディスプレイ数式モードは，￥[と￥]で括られています。

LaTeXでしか使えませんが，ディスプレイ数式モードでは，数式番号を自動的につけられます。これには，equation環境を使います。具体的には，1.3に書かれているとおりです。

数式番号をつけるということは，やはり大事な式というわけで，以後の章で参照される可能性があります。それには，￥label命令で数式にラベルをつけておき，￥ref命令で参照すれば可能です。そういう番号は覚えておいて，(1.3) 式では……とすればよいではないか？　といわれそうですね。でも，このように￥labelと￥refを使えば，途中で数式番号をつけたい式を追加しても以降の数式番号が繰り上がるので，多少タイプ数が増えたにしても，長いレポートや論文を書くとき格段に便利になるというわけです。

演算子などなど(印字例2)

それでは数式モードに入り，実際に数式を書いてみましょう。まずは小学校レベルの四則演算などから。

たとえば，4＋5×6－7÷8＝33.125という式を書きましょう。これは，2.1を見てください。+，－はそのまま書けばよいですが，×は￥times，÷は￥divという記号を使います。このような記号を利用するときは，必ず前後にスペースを1つずつ入れておくこと。TeXはほかにも±記号では￥pmといった具合に，数々のマクロをもっています。これらの記号一覧は，X68kPro

印字例1（69%に縮小）

1　数式モードの違い

TEX の数式は大きく分けて2種類あります．それらの違いを見てみましょう．

1.1　インライン数式モード

文中に数式を埋め込む数式モードです．たとえば、$i_c = \left(\frac{R_e+R_L}{R_e+R_L}\right) v_{CE} = \frac{v_{CE}}{R_{ac}}$ と、まぁこのような具合です．

1.2　ディスプレイ数式モード

別組みで数式を書くモードです．上の式は、別組みモードでは、このようになります．

$$i_c = \left(\frac{R_e+R_L}{R_e+R_L}\right) v_{CE} = \frac{v_{CE}}{R_{ac}}$$

こんな感じです．どうですか？

1.3　equation 環境

自動的に、数式番号をつけてくれる環境がこれです．

$$a = b = c \tag{1}$$

このような感じで、番号がつきます．ラベルをつけるためにはこうします．

$$f(t) = \sin(\omega t + \alpha \sin \psi t) \tag{2}$$

ラベルを張っておけば、あとは

(2) 式では……

このようにどこからも参照ができます．

リスト1

```
 1: ¥section{数式モードの違い}
 2:
 3: ¥TeX の数式は大きく分けて2種類あります。それらの違いを
 4: 見てみましょう。
 5:
 6: ¥subsection{インライン数式モード}
 7:
 8: 文中に数式を埋め込む数式モードです。たとえば、
 9: $i_c=¥left( ¥frac{R_e+R_L}{R_e+R_L}¥right) v_{CE}=¥frac{v_{CE}}{R_{ac}}$
10: と、まぁこのような具合です。
11:
12: ¥subsection{ディスプレイ数式モード}
13:
14: 別組みで数式を書くモードです。上の式は、
15: 別組みモードでは、このようになります。
16:
17: ¥[i_c=¥left( ¥frac{R_e+R_L}{R_e+R_L} ¥right) v_{CE}=¥frac{v_{CE}}{R_{ac}}¥]
18:
19: こんな感じです。どうですか?
20:
21: ¥subsection{equation環境}
22:
23: 自動的に、数式番号をつけてくれる環境がこれです。
24:
25: ¥begin{equation}
26:  a=b=c
27: ¥end{equation}
28:
29: このような感じで、番号がつきます。
30: ラベルをつけるためにはこうします。
31:
32: ¥begin{equation}
33:  f(t)= ¥sin ( ¥omega t + ¥alpha ¥sin ¥psi t)
34:    ¥label{eqn:a1}  % この行がラベル指定(%はコメント行ね)
35: ¥end{equation}
36:
37: ラベルを張っておけば、あとは
38:
39: ¥begin{center}
40:  (¥ref{eqn:a1})~式では$¥cdots ¥cdots$
41: ¥end{center}
42:
43: このようにどこからも参照ができます。
```

gramming Seriesの「X680x0 TeX」, Vol. 2の134ページをご覧ください。

四則演算の表記がわかったところで，次に出てくるのは分数です。2.2を見てください。このように，分数は¥frac{分子}{分母}という命令で表します。当然のことながら，分数のなかに，分数を入れることもできます。

また，ルートなどは，BASICの関数のように，やはり関数っぽく書きます。2.3の例のように，¥sqrt{中身}と書くのです。当然ながら，ルートのなかにルートや，分数などを書くこともできます。

なかに分数が入る場合，ルートのサイズが心配になりますが，いつものように，LaTeXでは，自動的に大きさを変えて処理してくれます。

添字の表記（印字例3）

次は添字。要は乗数などの表記ですね。

乗数はBASICと同じように，ax^2＋bx＋cといった感じで，「^」を利用します。乗数の乗数も，当然可能です。乗数に分数などを組み合わせることも，当然可能になります（3.1)。

下付きの添字はBASICなどの演算子にはありませんが，「_」(右シフトの左のキーをシフトと同時に押す）を利用します。また，ない文字に関しても，添字を書くことができます。たとえば$_nC_k$などは，{}_nC_kといったように書くわけです（3.2)。

注意しなくてはならないことは，「^」も「_」も，{}で括らないと，次に続くひとつしか添字にしないことです。たとえば，10^－3などとリストに書いたとき，10の上付きになるのは，－だけで，3は上付きにはなりません。

また，3.3のように上付きと下付きが同時についているものも書くことができますし，Σ記号はディスプレイ数式モードだと，教科書などにあるようにちゃんと付加することができます。

添字とはちょっと違いますが，レポート

印字例3（66%に縮小）

3　添字の表記

3.1　上付きの場合

乗数はBASICと同じように表記します。

$$ax^2 + bx + c = 0$$

以下は乗数の乗数の例です。

$$f(t) = \epsilon \cdot \sin(\omega t + \varepsilon^{\tan^{-1} \frac{\alpha^{\frac{1}{2}}}{\beta}})$$

ここまでくると、dpiがかなり高い出力デバイスでないと、わけわかんないですよね。

3.2　下付きの場合

$$i_{CE} = h_{FE} \times i_{BE}$$

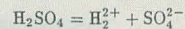
ちゃんと中括弧で括らないと、こんなになってしまいます。

$$i_C E = h_F E \times i_B E$$

こういう風に、ない文字に対する添字だってできます。

$${}_nC_k$$

3.3　上付きと下付きの場合

$$\mathrm{H_2SO_4 = H_2^{2+} + SO_4^{2-}}$$

$\sum_{n=0}^{\infty} An$ 文中ではこんな感じですが、別組みでは、

$$\sum_{n=0}^{\infty} An$$

このようになります。
当然こんな奴もあります。

$$\int_a^b f(x)g(x)dx$$

文中に入れると、こんなになってしまいますが$\cdots$ $\int_a^b f(x)g(x)dx$ まぁそうだろうなぁ。

3.4　ベクタの表現

$$\vec{a} \; \vec{ab}$$

リスト3

```
 1: ¥section{添字の表記}
 2:
 3: ¥subsection{上付きの場合}
 4:
 5: 乗数はBASICと同じように表記します。
 6:
 7: ¥[ax^2+bx+c=0¥]
 8:
 9: 以下は乗数の乗数の例です。
10:
11: ¥[ f(t)= ¥epsilon ¥cdot ¥sin ( ¥omega t + ¥varepsilon ^{¥ta
n ^{-1}  ¥frac{¥alpha^¥frac{1}{2} }{¥beta }}) ¥]
12:
13: ここまでくると、dpiがかなり高い出力デバイスでないと、わけわかんないですよね。
14:
15: ¥subsection{下付きの場合}
16:
17: ¥[i_{CE}=h_{FE} ¥times i_{BE} ¥]
18:
19: ちゃんと中括弧で括らないと、こんなになってしまいます。
20:
21: ¥[i_CE=h_FE ¥times i_BE ¥]
22:
23: こういう風に、ない文字に対する添字だってできます。
24:
25: ¥[ {}_nC_k ¥]
26:
27: ¥subsection{上付きと下付きの場合}
28:
29: ¥[{¥rm H_2SO_4=H_2^{2+}+SO_4^{2-}}¥]
30:
31: $¥sum_{n=0}^{¥infty}An$文中ではこんな感じですが、別組みでは、
32:
33: ¥[ ¥sum_{n=0}^{¥infty}An ¥]
34:
35: このようになります。
36:
37: 当然こんな奴もあります。
38:
39: ¥[¥int _a^b f(x)g(x) dx ¥]
40:
41: 文中に入れると、こんなになってしまいますが$¥cdots$ $¥int _a^b f(x)g(x) dx
$まぁそうだろうなぁ。
42:
43: ¥subsection{ベクタの表現}
44:
45: ¥[¥vec{a} ¥; ¥; ¥vec{ab} ¥]
```

などを書く際，変数は必ずしもスカラーではなく，ベクタになる場合があります。このようなベクタ表記（3.4）は，¥vec命令を使って表記します。この命令もほかと同じように，1文字なら，¥vec aでもよいのですが，2文字以上なら，¥vec{ab}というように中括弧で括ってください。もちろん1文字のものを中括弧で括ってもかまいません。

ベクタ記号のように，文字の上に飾りをつける記号は，ほかにもいろいろあります。これらの種類については，「X680x0 TeX」vol.2の139ページを参照してください。

括弧あれこれ（印字例4）

ちょっとでも複雑な数式を書くと必要になってくるのが括弧です。通常使うときには，普通に () とタイプすれば，括弧を書くことができます（4.1）。

しかしながら，4.1のいちばん外側の括弧のように，数式のサイズに合わせて，やっぱり大きくしたいときもあります。こういうときには，¥left(と¥right)を利用します。上の式をこの命令を使って書くと4.2のようになります。こちらのほうが美しいですよね？　タイプは面倒ですが。

括弧は () のほかにも，[]，{}，| |，|| || があります。{} などは，TeXでは機能的に使われているので，このような括弧は，無条件に¥left¥{，¥right¥}としなくてはなりません（4.3)。余談ですが，リストの¥；はスペースを意味します。このあたりはあとで説明しましょう。

4.4のように，両側で形が違ったり，片側がない括弧も作ることができます。括弧の話ではないのですが，例の「″」はリストにあるとおり，TeXでは「' '」と書けば自動的に処理してくれます。「‴」の場合でもちゃんと処理します。

括弧の極めつけは，上下の括弧です。これらは，¥overbrace,¥underbraceを利用します。例を見てください（4.5）

すごいことになっていますが，このようになります。この2つのコマンドは，

¥overbrace {括弧で括る範囲} ^ {上の注}

¥underbrace {括弧で括る範囲} _ {下の注}

印字例4（69%に縮小）

4　括弧あれこれ

4.1　普通に括弧

単なる () を使うと、このようになります。

$$(A \times \sqrt{B + (\frac{1}{3} + \frac{\sqrt{c}}{d})})^{-2}$$

4.2　括弧の大きさを自動的に…

自動的に変えるにはこのようにしなくてはなりません。

$$\left(A \times \sqrt{B + \left(\frac{1}{3} + \frac{\sqrt{c}}{d}\right)}\right)^{-2}$$

4.3　その他の括弧

$$\left[\sqrt{\frac{a}{b}}\right],\left\{\sqrt{\frac{a}{b}}\right\},\left|\sqrt{\frac{a}{b}}\right|,\left\|\sqrt{\frac{a}{b}}\right\|$$

4.4　両側で違う例

範囲の指定とかは、こうしますよね？

$$(x,y)=\left[0,6\right)$$

括弧の省略の例など。

$$\left.\frac{d^2y}{dx^2}\right|_{x=a} = f''(a)$$

4.5　上下の括弧

これは CRTC のレジスタ R00 のお話。

$$R00 = \mbox{水平同期期間} \div \bigl\{\overbrace{\mbox{水平データ表示期間} \div \underbrace{\left(\mbox{水平表示ドット数} \div 8\right)}_{\mbox{ドット数を8で割る}}}^{\mbox{8 dot clock を意味する}} \bigr\}-1$$

リスト4

```
 1: ¥section{括弧あれこれ}
 2:
 3: ¥subsection{普通に括弧}
 4:
 5: 単なる$()$を使うと、このようになります。
 6:
 7: ¥[ ( A ¥times ¥sqrt{ B + ( ¥frac{1}{3} + ¥frac{¥sqrt{c}}{d} ) } )^{-2} ¥]
 8:
 9: ¥subsection{括弧の大きさを自動的に$¥cdots$}
10:
11: 自動的に変えるにはこのようにしなくてはなりません。
12:
13: ¥[ ¥left( A ¥times ¥sqrt{ B + ¥left( ¥frac{1}{3} + ¥frac{¥sqrt{c}}{d} ¥righ
t) } ¥right)^{-2} ¥]
14:
15: ¥subsection{その他の括弧}
16:
17: ¥[ ¥left[  ¥sqrt{ ¥frac{a}{b}} ¥right]  ¥;,¥;
18:     ¥left¥{ ¥sqrt{ ¥frac{a}{b}} ¥right¥} ¥;,¥;
19:     ¥left|  ¥sqrt{ ¥frac{a}{b}} ¥right|  ¥;,¥;
20:     ¥left¥| ¥sqrt{ ¥frac{a}{b}} ¥right¥| ¥]
21:
22: ¥subsection{両側で違う例}
23:
24: 範囲の指定とかは、こうしますよね?
25:
26: ¥[ (x,y)=¥left[ 0,6 ¥right) ¥]
27:
28: 括弧の省略の例など。
29:
30: ¥[ ¥left. ¥frac{d^2y}{dx^2} ¥right|_{x=a} = f''(a) ¥]
31:
32: ¥subsection{上下の括弧}
33:
34: これはCRTCのレジスタR00のお話。
35:
36: ¥[ R00=¥mbox{水平同期期間} ¥div ¥bigl¥{¥overbrace{¥mbox{水平データ表示期間} ¥div ¥und
erbrace{¥left(¥mbox{水平表示ドット数} ¥div 8 ¥right) }_{ドット数を8で割る}}^{¥mbox{¥script
size 8 dot clockを意味する}} ¥bigl¥}-1  ¥]
```

とこのように使います。

新しいコマンドに¥mboxというものがありますが，これは式のなかに文章を入れるものです。日本語だけでなく，当然英語も使えます。また¥bigl，¥bigrというのは，さっきの¥left，¥rightのちょっと大きな括弧です。

しかし，ここまで長い式だと，発狂しそうですねぇ。どことどこがネストしているんだか，だんだんわからなくなってしまいます。

関数など(印字例5)

すでにいろいろなところで当然のように使っていましたから，あえていまさらいう必要はないでしょうが，関数は¥から始めるコマンドで表記します。

5ではsinについて3種類の書き方を行っています。そのままsinと書くと，フォントがイタリックになってしまい，変数との見分けがつきにくいため，関数はこのように¥から書きます。

印刷結果を見ればわかるとおり，¥sinは，¥mbox{sin}をそのまま定義していると考えれば，わかりやすいかもしれません。したがって，扱いとしては¥sinは1文字です。

同様に，ギリシャ文字，数学記号なども同じように¥から書きます。これも初めから使っているのでわかるでしょう。たとえばαを表示したいならば¥alpha，Γを表示したいならば¥Gammaといった感じです。

印字例5（69%に縮小）

5 関数

このような違いがあります。

$$f(t) = sin(\omega t) = \sin(\omega t) = \mathrm{sin}(\omega t)$$

リスト5

```
 1: ¥section{関数}
 2:
 3: このような違いがあります。
 4:
 5: ¥[ f(t)=sin(¥omega t) = ¥sin (¥omega t) = ¥mbox{sin}(¥omega
t)¥]
```

印字例6（69%に縮小）

6 行列

6.1 簡単な行列

$$\left(\begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array}\right)$$

6.2 行列のなかに行列がある例

$$\left|\begin{array}{cc} \left|\begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array}\right| & \mbox{B} \\ \mbox{C} & \left|\begin{array}{cc} d_{11} & d_{12} \\ d_{21} & d_{22} \end{array}\right| \end{array}\right|$$

6.3 その他

当然こんなものもあります。

$$\left|\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} \end{array}\right|$$

また、行列とは違いますが、こういうものもあります。

$$x \choose y$$

行列を利用して、場合分け。

$$f(x) = \left\{\begin{array}{cr} 0 & \ldots \;\;\; x \neq 0 \mbox{の時} \\ \infty & \ldots \;\;\; x = 0 \mbox{の時} \end{array}\right.$$

リスト6

```
 1: ¥section{行列}
 2:
 3: ¥subsection{簡単な行列}
 4:
 5: ¥[
 6:  ¥left(
 7:   ¥begin{array}{cc}
 8:    a_{11} & a_{12} ¥¥
 9:    a_{21} & a_{22}
10:   ¥end{array}
11:  ¥right)
12: ¥]
13:
14: ¥subsection{行列のなかに行列がある例}
15:
16: ¥[
17:  ¥left|                       %  親の括弧(左)括弧は|も使うよね?
18:   ¥begin{array}{cc}           %  親の行列指定
19:    ¥left|
20:     ¥begin{array}{cc}         %  子の括弧その1
21:      a_{11} & a_{12} ¥¥
22:      a_{21} & a_{22}
23:     ¥end{array}
24:    ¥right| & ¥mbox{¥LARGE  B} ¥¥          %  この&Bは親のB
25:    ¥mbox{¥LARGE  C} & ¥left|              %  このA&は親のA
26:     ¥begin{array}{cc}         %  子の括弧その2
27:      d_{11} & d_{12} ¥¥
28:      d_{21} & d_{22}
29:     ¥end{array}
30:    ¥right|
31:   ¥end{array}                 %  親の括弧のネスト
32:  ¥right|                      %  親の括弧(右)
33: ¥]
34:
35: ¥subsection{その他}
36:
37: 当然こんなものもあります。
38:
39: ¥[
40:  ¥left|
41:   ¥begin{array}{cccc}
42:    a_{11} & a_{12} & ¥cdots & a_{1j} ¥¥
43:    a_{21} & a_{22} & ¥cdots & a_{2j} ¥¥
44:    ¥vdots & ¥vdots & ¥ddots & ¥vdots ¥¥
45:    a_{i1} & a_{i2} & ¥cdots & a_{ij}
46:   ¥end{array}
47:  ¥right|
48: ¥]
49:
50: また、行列とは違いますが、こういうものもあります。
51:
52: ¥[ x ¥choose y ¥]
53:
54: 行列を利用して、場合分け。
55:
56: ¥[
57:  f(x)=
58:  ¥left¥{
59:   ¥begin{array}{cr}
60:       0   & ¥ldots ¥;¥;¥; ¥mbox{$x ¥neq 0$の時} ¥¥
61:    ¥infty & ¥ldots ¥;¥;¥; ¥mbox{$x   =  0$の時}
62:   ¥end{array}
63:  ¥right.           % これがポイントかな?
64: ¥]
```

今月のリストについて

今月は1本のリストを各セクションごとに分けた。印字例で1〜7のセクションがあるのはそういうわけである。したがって各リストをそれぞれコンパイルすることはできない。念のため。

あと，数式モードに入る方法をインライン数式が＄〜＄，ディスプレイ数式が¥[〜¥]で統一した。数式モードに入るための書式はほかにもあるが，各自で「X680x0 TeX」などを参照してほしい。

ギリシャ文字を大文字で書く場合，このようにスペルの最初を大文字にするのです。

すべて，１文字として扱われるので注意してください。

行列(印字例6)

そして行列です。

行列には複雑怪奇なものがいろいろありますが，ここでは，簡単なものだけ扱うことにします。

まず，普通の２次行列です(6)。ディスプレイ数式モードですから，¥[，¥]間でネストされます。そして，¥left(，¥right)です。()だけでは，サイズを合わせてくれないので，これを利用します。

そして，array環境です。これは前回のtabular環境と法則は同じで，次の {cc} が行の数，センタリングを意味します。

行列のなかに当然，行列も書けます(6.2)。リスト６のコメント (%以降) を参考にしてください。基本的な考え方は，親全体から見れば，子の行列環境，¥begin{array} ～¥end {array} までは１要素と考えます。

6.3は，当然こんなものも書けるという例です。「…」である¥cdots以外に，斜めの¥ddots，縦の¥vdotsといくつかあるといったところですか。

ほかにも，行列コマンドと，括弧を表示しない¥right.を利用して，数式の場合分けを行ったりできます。

これ以上複雑怪奇な行列を書く方は，専門書をお読みください。

その他(印字例7)

TeXが数式モードで自動的に整形してくれるものの，ときどきもう少し空白がほしいというときがあります。

たとえば，7.1の最初の式の左辺のように括弧とAが密着してしまい，少しだけ空白がほしくなることがあります。右辺では，¥；で空白を作っています。

空白コマンドはほかにもあり，¥,，¥：，¥；の順で広くなっていき，¥！で負の空白を空けることができるので，7.1の最後の式のように詰めることもできます。

そして，ここで登場した，eqnarray環境。&～&で括られた文字列を基準に連続した式を揃える環境です。¥¥は改行を意味します。ここで利用したものは，eqnarrayのあとに＊マークがついているので，この場合，数式に番号を振りません。

普通にeqnarray環境を使うと，7.2のように，数式に番号がついてしまいます。式の変更まで書かなければいけないレポートなどでは，どちらかといえば，¥[～¥]でネストされた数式環境よりも，こちらのほうが使われるでしょう。

当然ながら，¥label，¥refの利用もできますし，式の途中なので番号を振りたくないという場合には，¥nonumberを付記して，回避することができます。

印字例7（69%に縮小）

7　その他

7.1　空白がほしい

$$A(3+5) = A\;(3+5)$$

空白の例

$$\begin{aligned} B &= A(3+5) \\ B &= A\,(3+5) \\ B &= A\:(3+5) \\ B &= A\;(3+5) \\ B &= A\!(3+5) \end{aligned}$$

7.2　eqnarray 環境

数式に番号がつく、eqnarray 環境。

$$\begin{aligned} f(x) &= 3\times 4+6\div 2 \\ &= 12+3 \\ &= 15 \qquad (3) \end{aligned}$$

当然、(3) 式ではと引用できます。

リスト7

```
 1: ¥section{その他}
 2:
 3: ¥subsection{空白がほしい}
 4:
 5: ¥[ A(3+5)=A ¥; (3+5) ¥]
 6: %             ^^ これが空白
 7:
 8: 空白の例
 9:
10: ¥begin{eqnarray*}
11:  B&=&A(3+5) ¥¥
12:  B&=&A¥, (3+5) ¥¥
13:  B&=&A¥: (3+5) ¥¥
14:  B&=&A¥; (3+5) ¥¥
15:  B&=&A¥! (3+5)
16: ¥end{eqnarray*}
17:
18: ¥subsection{eqnarray環境}
19:
20: 数式に番号がつく、eqnarray環境。
21:
22: ¥begin{eqnarray}
23:  f(x) &=& 3 ¥times 4 + 6 ¥div 2 ¥nonumber ¥¥
24:       &=& 12 + 3 ¥nonumber ¥¥
25:       &=& 15 ¥label{eqn:a2}
26: ¥end{eqnarray}
27:
28: 当然、(¥ref{eqn:a2})~式ではと引用できます。
29:
30: ¥end{document}
```

まとめ

以上，数式に関してはだいたい終わりです。BASICやCなどでの数式の入力に近いものがあるので，たぶん，プログラミング言語を一度でも使ったことがある人なら，すぐにTeXに親しめると思います。

今回説明した以外にも，数式のフォントを変えたり，括弧を好きな大きさにしたり，いろいろな機能があります。これらについては，専門書を参考してください。

さて，来月は，表の作り方です。１回目に表をちょこっとだけ作りましたが，これらをさらに応用するテクニックを書きたいと思います。

参考文献：第２版楽々LaTeX　野寺隆志著
共立出版株式会社
注意：第２版と第１版は大違いです。
できるだけ，第２版を購入するように。

第86回　知能機械概論──お茶目な計算機たち──

インタラクティブ・エボリューション

究極のマシンで

計算機のスピードは速くなる一方ですが，このところスピードの上昇度はゆるやかになってきたかもしれません。確かに計算機アーキテクチャはこの10年くらいの間，あまり革新的な変化はとげていないといってよいでしょう。もしかして，もうすぐ計算機のスピードは頭打ちになってしまうのでしょうか？

もちろん，答えは「NO」です。将来は半導体素子の代わりに超伝導だ，光だといってもいいのですが，それより前に，アーキテクチャ上の大きな革新が期待できると思います。それが超並列マシンです。

超並列マシンは，プロセッサを多数個接続して構成するものです。単にプロセッサをたくさんつなぐだけならば，理論的にそう難しい問題ではないでしょう。

しかし，問題はソフトウェアです。計算機に与える問題の記述法や，コンパイル，分割してプロセッサに与える方法などをクリアしていかねばならないのです。

ここでは，この大難問は他人ごとのようにあっさりと放っておくことにします。次に「究極的に速い計算機ができあがりました。さて，この計算機になにをさせたら面白いでしょうか？」という質問に答えてみましょう。いってみれば，人工生命の味付けをふんだんにきかせた夢物語です。

壮大なドラマ

計算機のスピードが速くなるということは，計算機内部の仮想世界に存在する時計の針の回転速度が速くなるということを意味します。究極的な超高速マシンの中では，あっという間に時間が過ぎていきます。

ビデオカメラを固定して1日中撮り続けたテープを高速に再生するとひまわりの花がけなげに太陽を追いかけている姿を見ることができるように，時間軸を変えると見えていたものが見えなくなったり，見えなかったなにかが見えるようになります。

超高速マシンで世の中を再生してみるとなにが見えてくるのでしょうか？　宇宙や惑星のダイナミックなドラマを見ることができるでしょう。また，視点を地球上に合わせるとさまざまな生物種のドラマを見ることができることでしょう。これが進化のドラマです。

超高速マシンでしか見ることのできないこのようなドラマ，それは従来のことばでいえばシミュレーションということになるのでしょうが，格別に面白いぞと僕は考えています。

「インタラクティブ・エボリューション」

適当なことばを思いつかなかったので，シミュレーションということばを用いてしまいましたが，あまり適当ではないかもしれません。なぜならば，通常シミュレートするということばは，現実のなにかをまねしてみるという意味で使われるからです。

確かに，現実の世の中を再現するという一面はあります。そして，運命論者ならば，「シミュレーションそのもの」というかもしれません。しかし，「いまの状態はたまたまこうなったというひとつの結果にすぎない」という側面が多かれ少なかれあるでしょう。

そう考えたうえで，計算機の中で昔のある時点からシミュレーションを開始したとします。すると，1回試したくらいでは，現実の世の中に似た状態になったり，あるいは我々のような人類が誕生する可能性はほとんどありえないということになります。現実の歴史をなぞっているのではないという意味で，シミュレーションということばは適当とはいえないということです。

ここで説明しようとしている概念を表すことばを考えました。それは，「インタラクティブ・エボリューション」(Interactive Evolution：相互に作用する進化)です。通常意味される「進化」は，根から種が出発してときどき枝分かれして進んでいくわけですが，インタラクティブ・エボリューションでは，そのような単調な方向性をうち破ります。進化させていくどんな時点においても，そこで時間の流れを止め，また過去のある時点から再開することができるのです。

しかも，同じ条件で始めたとしても，場合によってはまったく違う進化の流れにのっていることもしばしばあるわけです。実はこのことが単なるシミュレーションにとどまらない興味深い点でもあります。ある事柄が必然的ではなくたまたまそうなったということもわかってきます。

進化の流れにのっていることには違いないのですが，進化の進み具合を見ながら，嫌ならばまた過去のどの時点からもやり直しがきくという，きわめて都合のよい歴史です。そういう意味を表すことばとして，流行のことばのようですがインタラクティブとつけました。

アミューズメントとしての性格

すべてのパソコンゲームはシミュレーションであるということができます。どんなゲームでもある特定の状況を想定したうえで，その状況を計算機内の0や1の列によって，シミュレートしていることには違いないからです。

そして，インタラクティブ・エボリューションの基本的な性格はアミューズメントであるともいえます。誰もが楽しめる遊園地のようなイメージでしょう。何が出てくるかまったくわからない夢の国です。しかも，参加する人はネットワークにより世界を共有しています。

しかし，ある進化系列上で，ある一時点のある一地点に自分はいるわけですから，自分以外の人は基本的には仮想的な生物であると考えてよいでしょう。むろん，特定の進化系列上でのある場所と時刻を決めれば，友人と待ち合わせることもできます。

インタラクティブ・エボリューションの実現にはまだまだ時間が必要でしょう。理論的に固めなければならないことが山積みにされています。しかし，このシステムが本質的にもつ誰もがのめりこむようなアミューズメント性は，システムが未完成であっても，人々をひきつけ，産業としても拡大していく起動力となるでしょう。

新しい知能との共存

楽観的に考えれば（ダーウィンにその源をおく自然淘汰や突然変異などの原理以外に，埋め込むべき種々の原理が今後明らかにされればという意味です），計算機の中でインタラクティブに種を進化させることにより，人間の知能を超えるような新たな知能というものも自然に発生させることができるでしょう。

それは人類がさらに進化したものか，あるいはもっと根元のほうからいまの人類とは異なったものに枝分かれした結果生まれたものかはわかりませんが，そのような知能というものは，もはや人類の頭では理解することができないでしょう，我々の知能を超越しているのですから。

しかし，少し考えてみれば，インタラクティブ・エボリューションということばで示されるシステム自体がすでに，新しい知能そのものであると考えられると思います。それは2つの意味においてです。

知能とはなにかと考えたときに，シミュレーションする能力そのものだと定義することができます。「これをするとこうなってこうなるから駄目だ」とか「ああすればああなるからあれをやったほうがよい」などとできることがすなわち知能であるということです。そういう意味でいえば，インタラクティブ・エボリューションとはまさにその定義そのものとなります。

また，インタラクティブ・エボリューションとは時間や空間という制約，物理的束縛を断ち切った世界を想定して，自由に進化を行わせるものですから，実はそこで人類の次に生まれ出た知能体は，どんなものであろうとも進化系列において人間の次に来るものという意味において新しい知能そのものにまちがいないということです。

人類は別になにも恐れることはありません。それどころか大変に頼もしい仲間の登場を喜ぶべきです。この環境のなかで生きるために，我々は現在だってアメーバや細菌類などとも共存共生しているわけですし，そこに強力な助っ人が登場してくれるというわけですから。しかも，実際の進化の歩みの遅さをがまんして待つことなしに大急ぎでやってきてくれるのです。

この共存している状態を数学ふうに表現するならば，ひとつの大きな問題を部分問題に分け，それぞれを人類や新知能体を含む各種の生物が分担して解いているといったところでしょうか。

illustration：Haruhisa Yamada

取り扱い注意

日本では進化という概念に興味をもっている人も多く，比較的すんなり受け入れられるようですが，キリスト教が比較的勢力をもっている地域では，このような進化論をベースとする話をすると苦労するかもしれません。

たとえば，アメリカのルイジアナ州（少年を犬猫のように撃っても無罪になったところ）には，公立の小・中学校では，進化論と天地創造を同じ時間だけ教えなければいけないという，僕にとってはまったく信じられない州法があるそうです。

矛盾する2つの話を同時に教えられた小・中学生はどのように受け止めるでしょうか？　より深い洞察力をもつように鍛えられたり，論理的な考え方と宗教的な考え方の切り分け方を学べばよいのですが。

また，進化論的な考えは俗っぽい話や危険な話に結びつきやすい側面ももっています。世の中の一時的な風俗現象，社会現象を種の進化という話に根拠もなしに安直に結びつけ，結果的には自分の意図する一定の方向にもっていこうとする本がよく見られます。しかも，よく売れるのです。

進化論的な考えというのは，ダイレクトに良い/悪いの価値判断につながるものではありません。もしなんらかの尺度として用いる際にも，ほかのさまざまな基準とのバランスがどうしても必要なのです。極端な話，「～のような人は人類の存続にとって望ましくないので子供を作らないほうがよい」などというような暴論の土台として使われてしまうからです。

そして人類は悟る

インタラクティブ・エボリューションが人類に与える影響のうち，精神的な面へのインパクトは計りしれないものがあると思います。毛利さんや向井さんの場合は，宇宙が素晴らしいというイメージ以外のなにかをまだ僕には教えてくれないようですが，過去の宇宙飛行士の多くは悟りというか，人生観が変わるようななにかを受けたといっています。なかには本当に宗教家になった人もいましたね。

人間はもともと自分の周囲のせいぜい数百メートル以内の範囲で活動するように体も頭もできているのでしょう。その活動空間を一気に宇宙にまで広げることにより，宇宙飛行士の頭脳はその思考の枠を大きく広げ深い精神空間をもつようになるのだと思います。

時間に関しても同様でしょう。10分後，明日，あるいはせいぜい5年後までを考えるのに慣れている我々は，時間軸を自由に行き来することにより，とてつもない変貌を遂げるかもしれません。たとえば，ミクロなレベルの事象にとらわれない，より深い精神性を獲得するかもしれません。きっと，そんな安っぽいことばでは表せないような「悟り」が起こるのでしょう。

そのときこそ，我々が真の意味での「新人類」になるときかもしれません。

第96回

猫とコンピュータ

幻のディレクトリ

Takazawa Kyoko
高沢 恭子

目で見えるものこそ最も確かな真実のように感じられるものですが，ときどき，騙されてしまうこともあるようです。見えない裏側には，思わぬ幸運や困った事態などが待ち伏せしているかもしれません。

新入りのDOS/Vマシンは，先輩であるプリンタとの仲が悪い。どちらがどちらを気に入らないのか，印刷をしない。

DOS/V専門業のY氏はプリンタを自分のショップに持ち帰り，ほかのマシンとつなげるなどテストをしてくれた。結果はプリンタには問題ないとわかった。そこでこんどは新入りマシンが，精密検査のために入院となった。最悪の場合はCPUの交換だそうだ。

霧は深いほうがいい

鈴鹿山脈の尾根は三重県と滋賀県の県境の一部である。その最高峰が御在所岳で，鈴をつけた鹿の神が住む山という伝説があるそうだ。

DOS/Vマシンは入院中。夫はお盆休み。標高1,210メートル，国定公園でもある御在所岳のロープウェイからの眺望は格別と聞く。8月14日，探訪をかねたドライブに出かけた。

ふもとの湯の山温泉までクルマ，ここから全長2,200メートルの話題のロープウェイがはじまる。ゴンドラは38両が巡回式に1分ごとに発車，定員10名で約12分かかって山頂の駅までいく。

待ちながら見上げる御在所岳が白く霞んで見えるのが気になった。乗る人と同じ数だけ下車する人たちが，傘を持ったりビニールコートを着たりしている。

8人の人たちと乗り合わせてゴンドラは発車した。温泉の町や木々の群が足元を遠ざかるのはとてもスリリングだ。だが1〜2分のうちに私たちは白いまぶしいものにすっぽり包まれた。深い霧だった。ガラスをぬぐっても何も見えない中で，ゴンドラが上へ移動していることだけがわかる。下降のゴンドラが時折，白いミルクの中に浮かんでは消えていく。不安なような，愉快なような時間だったが，山頂はもっと幻想的だった。

すべては霧の中，視界は数メートルしかない。白いガスのような霧の中から，人影が夢のようにあらわれる。映画やテレビの幻想シーンは，たいへんな準備をしての撮影かと思うが，このままの実写でいいじゃないかなんて思った。

行きかう人はぐっしょり濡れている。下の駅で見た雨具は霧よけだったのだ。カモシカセンターを見学，昼食のあと下山のための順番待ちをしながら，霧が深かったことを私は残念に思っていた。

下界に向かうゴンドラははじめから何も見えなかったので，「晴れた日と同じ料金はヒドイんじゃない？」とついにグチも出た。白く明るく濁った空中を，ただ，ハコが動いていく。

一瞬！　風で動いた霧の裂け目から谷が見えた。その鋭い深さ，緑のあざやかさに思わず声をあげた。それはすぐに消えた。また数秒後に，光る鏡のように谷底があらわれて，消えた。なんて美しい凄い眺めだろう。いままでに見たこともない。そんなきらめく瞬間を，帰路のゴンドラでは何回か味わうことができた。

「霧の山はなかなか経験できないものだよ。かえってよかったんじゃないかな」と夫がいった。視界をふさぐ霧があったから，あの樹木の輝きが見えたのだ。

翌15日は夕刻から春日大社と東大寺をめぐった。双方とも境内のすべての灯籠に火をともす送り火の祭で，それぞれ，「万灯籠」「万灯供養」という。

ようやく暗くなりはじめた午後7時ころからいっせいに灯籠に火が入り，待ちかねたたくさんの人たちが参拝の歩みをはじめる。明治神宮の初詣のような人の波は，蒸し暑さも加わって，ちょっとしたガマンのお祭でもある。

春日大社の長い参道に沿った灯の列。東大寺の本堂を浮かび上がらせた華やかな光の重層と読経のライブ。オレンジ色の光がいまもゆらめいている。

ライオンのシンバ君

お盆が済むのを待って帰京した。用事やしごとがだいぶたまっていた。ディズニーの「ライオンキング」を見ようというのもトオルとの約束だったので，2人で有楽町まで行った。

「ライオンキング」は6月に全米で公開されると同時に爆発的なヒットとなったそうだが，手塚治虫の「ジャングル大帝」との類似性で，日米のアニメ関係者や劇画作家たちの論議の的となっている。

しかし，実際に見てみると，「ライオンキング」は，そういう批判の打撃をあまり受けそうにない，力強い作品だった。主人公も主要な脇役の動物たちも，みな厳しさのあるキャラクターが強調されていて，手塚アニメとは別の辛口の味が印象に残った。あらゆる設定がそっくりといわれているけれど，作品を見た人は誰もその力量に圧倒されたはずだ。

ディズニー側がいろいろと指摘されている中に，主人公の名前のことがあった。「ジャングル大帝」が1966年に米国版として放映されたときの主人公が「キンバ」であり，「ライオンキング」は「シンバ」である，ということだ。

「ヘンだなぁ」と，トオルがいった。

「ボクたちは小学校のころから『ライオンのシンバ君』とか，『シンバ君のライオンカ

レー』なんていう本を，学校の図書館で見てるよ」

トオルはついでに,「おもしろいものがある」といって，8年前に友人の石沢君が作って歌った曲「ライオンのシンバ君」のテープを聞かせてくれた。

ディズニープロが「ライオンキング」の制作をはじめたのは4年前だ。「シンバ」の名前はそれ以前から複数の本に使われているのだから，すくなくともディズニープロが今回のアニメのために「キンバ」をマネてつけたものではない。「シンバ」は何かライオンにかかわる由来があるのではないか。これがトオルの意見だ。

サークル活動のついでに大学の図書館で調べようとしたが休暇中で閉館だったといって，トオルは区内の図書館に行った。

「わかったよ！　『シンバの国ー羽仁進のアフリカ動物画集』にあった。『シンバ』はアフリカのスワヒリ語で『ライオン』のことだったよ」

トオルの推測どおりだった。そうしてみると,むしろ過去に米国版「ジャングル大帝」の主人公を命名したときの「キンバ」が「シンバ」を模していたといえるのではないか。

果てしない階層の怪

幻のようなもの，幻を思わせるものは多い。霧の中の樹木や人影は，現実であるのに幻影のような演出をする。灯籠や行燈の明かりも幽玄の世界を思わせる。

逆に，映像やアニメーションは，幻影であるのに現実のような世界を現出させて，私たちの想像の空間を広げてくれる。

この夏，わが家ではパソコンの中に幻があらわれた。問題の主は，夫が使っていた3.5インチのディスクである。

とりあえず何でも記録するディスク，いちばん身近なディスクでもある。各ネットのログ，会社でのロータスのファイル，会議の議事録。雑記帳のようなディスクの中には，わが家の貯蓄のメモの一部もまじっている。ファイルはすぐに数がふえるので，必要に応じてディレクトリをつくり，分類，整理をしていく。

ファイル数が49にもなったので，夫はディレクトリをつくって，ひとまとめにしようとした。「MD　倉庫」。ファイル名を漢字にするのはやめようかなと，そのときチラッと思ったという。DIRで「倉庫」のディレクトリができたのを確認した。つぎに「COPY　*.*　¥倉庫」で全部のファイルを「倉庫」の階層内にコピーした。コピーをしたあと，最上階のディレクトリのファイルをすべて消去するつもりだった。

無事にコピーが完了したことを確認するために，いったん「DIR　¥倉庫」をしてみた。そこで，CRTに思いがけない内容を見つけたのだ。

illustration : Kyoko Takazawa

コピーした49のファイル名だけが並ぶはずの画面に，おまけがついている。なんと,「倉庫<DIR>」が先頭になって，そのあとに49のファイル名が並んでいるのである。

つまり，いちばん上の階層で単にDIRをしたときと，まったくおなじ内容の画面表示なのだ。

おかしい……。確認のために,「CD　倉庫」をして「倉庫」の階層に入り，DIRをしてみる。やはり「倉庫　<DIR>」が先頭になって49のファイルが並ぶ。

どうしたのか。それならいったん，DELをしてみよう。「DEL　*.*」で「倉庫」の階層の中をすべて消去した。

「CD ..」でルートディレクトリに上がり，確認のためにDIRをした。

ない！　全部消えていて何もない。

倉庫の階層の中を消した操作で，上の階層まで消えている。

あわてて，また倉庫の階層に入ると，ちゃんとプロンプトは「A:¥倉庫>」となる。そこで DIRをすると，ファイルはないが「倉庫　<DIR>」がやっぱりある。

さらに「CD 倉庫」とすると今度はプロンプトは「A:¥倉庫¥倉庫>」となる。DIRをすれば，また「倉庫　<DIR>」がある。この階層はいくらでも続くことがわかった。

ディレクトリは作られたように見えて，じつは正しいディレクトリはできていなかった。「CD　倉庫」のコマンドで入ったつもりの階層は，虚構だったのだ。

上の階層のファイル名が，まるで鏡にうつったように，ニセの階層の中に見えたのだろうか。DELコマンドで両方が消えてしまったのは，そうした鏡のトリックだったのだろうか。

「¥倉庫」は，けっきょく幻のディレクトリだった。なぜこんなことになったか。

ディスクを雑記帳がわりに使ったため，書いたり消したりの繰りかえしで，クラスタへのファイルの入力がバラバラになりすぎた。そこへ漢字のディレクトリ名を使ったことなどから，FATが破壊されたのではないか。

「DOSのバグかもしれない」と夫はいう。

どうもそのような使い方をしたときは，ファイルが破壊される。そして，あたらしいバージョンになったときに修正されたという話を聞いたことがあるそうだ。

とても暑い夏だった。DOSだって夏バテの空振りをすることもある。

こちらシステムX 探偵事務所

FILE-XVI

illustration : T.Takahashi

プログラミングを考える

Shibata Atsushi 柴田　淳

今月はプログラミングに関するお話です。今回のピンボール作成で利用したプログラミング方法について簡単な解説をしていきます。気になるピンボールの制作ですが，台のデザインもほとんど終わり完成に近づいてきました。

日本で遊べるピンボールの台はすべてがアメリカ製であると思われる。アメリカのピンボールメーカーはおそらく4社で，そのうちDATA EASTの台以外は輸入業者が日本で代行販売しているらしい（DATA EASTの台も，日本法人が請け負っているのは販売だけなので似たようなものだが）。そのなかに，TAITOが代行販売しているWilliamsというピンボールメーカーがある。

ピンボール台には，よく見てみるとメーカーごとに色のようなものがある。DATA EASTの台のデザインなどは，どちらかといえば線の細い印象を受ける。一方，Williamsの作るピンボールは重厚なイメージをかもし出している。ピンボール台上部の両側は，単に絵柄で埋められているだけではなく，樹脂製の構造物が置かれていたりする。また台の構成も立体的だ。仕掛けの数が多いとかいうのではなく，ゲームに関係ない細部にまで非常に凝った造りがなされているのである。プラスティックのパーツひとつ作るにも，型を新たに起こしたりと馬鹿にならないお金がかかるはずなのだ。このあたり，きっとデザイナーはこだわりをもっているのだろう。

Williamsの最近のこだわり，というか傾向として，幅広の台（ワイド台などと呼ばれる）の採用がある。幅が広い分，仕掛けも多くなるし，ボールの動きもダイナミックになるのだ。特に「Star Treck the Next Generation」の目まぐるしいほどの立体ランプ，「I'm the Law」の高くそびえ立つリターンレーンなど，Williamsの台には目を引いてなおかつ面白いフィーチャーが盛りだくさんである。いま，Williamsのピンボールはけっこうアツい。

ところで，最近のWilliamsの台はやたらに重い。重すぎて「揺らし」がきかないほどなのだ。アメリカ人にはあれくらいがちょうどいいのだろうか。

関数名を参照する

今月は少し趣向を変えて，プログラミング寄りの話をする。本題に入る前に，まずはいくつか実験をしてみたいと思う。

僕がCを始めて間もないころ，関数と同じ名前のグローバル変数をとってコンパイラにしかられたことが何度かあった。X-BASICでは同様のことをしても一向に問題にならないのに，Cではどうしていけないのだろう，とちょっと疑問に思ったものだが，そういうものなのだろうと自分を納得させて，その場は特に深く考えることをしなかった。

ちなみに，GCCで関数名と同一のグローバル変数を定義する（リスト1参照）と，

～が別の宣言をされました

とエラーが表示される。確かに同一名を2カ所で宣言しているわけだが，ではどうしてこうすることがいけないのだろうか。変数と関数ではコード中で扱われ方が異なるのだから，このあたりをコンパイラ側で認識して，X-BASICがやっているように同名でも別物として扱ってくれてもよさそうなものだ（「そういう仕様なんです」といわれればそれまでだが）。

では一方で，関数と同一名のローカル変数をとるとどうなるだろうか。リスト1をコンパイルしてみると，

関数でないオブジェクトを呼ぼうとしています

というエラー表示が出る。ちなみにオブジェクトとは，値を格納，参照できるメモリ領域のことで，具体的には変数や構造体の

リスト1

```
 1: /* --関数と同名の変数をとる-- */
 2:
 3: #include"stdio.h"
 4:
 5: int      func1;
 6:
 7: void     main()
 8: {
 9:          int func2;
10:          func2();
11:          /* 関数でないオブジェクトを
12:          呼ぼうとしていると解釈される */
13: }
14:
15: int      func1(void)
16: /* 二重定義になる */
17: {
18:          printf("This is func1.¥n");
19: }
20:
21: int      func2(void)
22: /* 定義だけはされるが,
23:     mainからは呼び出されない */
24: {
25:          printf("This is func2.¥n");
26: }
```

リスト2

```
 1: /* --関数に値を代入する-- */
 2:
 3: #include"stdio.h"
 4:
 5: int func(void)
 6: {
 7:          printf("This is func.¥n");
 8: }
 9:
10: void main()
11: {
12:          func = 0;
13:          /* 不正な代入,
14:             つまり，関数名に値は代入できない*/
15: }
```

ことと思っておけばいい。つまり，「変数なんか呼べるわけねーだろ」というのがこのエラーメッセージの趣旨なのだ。が，変数をあえて呼び出そうとする人はまずいないわけで，ここでも，どうして同名の変数と関数を別物として扱ってくれないの？　という疑問は残る。

さてここで，ちょっと変わった実験をしてみよう。関数名と変数名がコンパイラによって別物として扱われないのなら，関数名を変数とし，値を代入するとどうなるだろうか。これもリスト2をコンパイルしてみると，

代入に不正な左辺値があります

としかられる。要するに，関数名に値を代入することは禁止されているらしい。これを突き詰めるとやっぱり，リスト中の識別子が関数名として定義されているか変数として定義されているかを，コンパイラ側ではきちんと把握している，ということになる。

謎は深まるばかりだ。こうなったらもうやけくそである。関数名に値を代入するのと逆に，関数名を参照してみたらどうなるか。恐る恐るリストをコンパイラに通すと，エラーなしでコンパイルされ，おまけに作られた実行ファイルも問題なく動く。関数は値を返すものだという話は聞いたことがあるが，関数名自体に値があるとはどういうことか。

ちなみにものの本によると，関数名というのはその関数へのポインタなのだそうだ。したがって，関数名を右辺として変数に代入すると，変数には関数のアドレスが代入される。代入は許されていないものの，関数のアドレスを参照はできるので，やはり関数と同名の変数を定義されると困る，ということのようだ。

引数の変数を呼び出す

ところで，関数名を左辺としてとることで関数のアドレスが得られるとわかると，こうして得られたアドレスを使って関数を呼び出してみたくなる。しかし，ただ呼び出すのではつまらないので，関数の引数と仕手関数のアドレスを渡し，関数アドレスを渡された先の関数でお目当ての関数を呼び出してみよう。

リスト3を見てほしい。このリストでは関数funcAのポインタ(つまり関数名)をfuncBの引数として渡し，funcBで渡された関数を呼び出している。関数funcAの定義がmainの上にあるのは，main中でfuncAのアドレスを参照する時点でfuncAが既に定義されている必要があるからだ。funcAを関数mainの下で定義する場合，プロトタイプ宣言をする必要が出てくる。

さて，このリストのミソは3行目の部分である。ここではtypedefを使って「void(空)の関数型」を定義している。関数を引数として受けるfuncBの定義部では，引数の型として3行目で定義された型をとるようにする。こうしないで，たとえば引数をlong型に取るなどとすると，コンパイラに変数を呼び出しているものと受けとられ，しかられてしまう。

関数funcBは，受け取った関数に99という数字を渡し呼び出す。funcBから渡された引数は当然funcAが受け取り，画面に99が表示されればこのソースは意図したとおりに動いていることになる。で，リスト3をコンパイル，実行してみると，ちゃんとそうなっていることがわかるはずだ。僕のCプログラミングの経験が浅いせいか，このリストはずいぶんと怪しく見えるのだが，問題なく動いてしまうので非常に不思議な感じがする。

ちなみに，関数main内の12行目を，

　funcB (&funcA)；

と書き換え，15行目を，

　void　　funcB (vfunc ＊f)

と書き換えてみる。つまり，関数へのポインタのポインタを渡し，funcBでそれを受けるようにするとどうなるだろうか。これがまた不思議なことに，問題なく動いてしまう。昔のコンパイラでは，このようなソースを書くと17行目を，

　(＊f) (99)；

としなければならなかったようだが，最近のコンパイラでは関数として定義されている識別子はその名前を書くだけで呼び出してくれる。

また，関数funcBの引数をlongとし，関数を呼び出すときにキャストで引数を関数型に変換するとどうなるか，と思って試してみたところ，こちらはコンパイラに「関数でないオブジェクトを呼ぼうとしている」と解釈されてしまった。変数をやたらと関数化されては困る，ということなのかもしれない。

データ構造に関数を埋め込む

僕はプログラムを書く以上，できるだけスマートなリストを書きたいと思っている。スマートであろうとなかろうと外見的には同じに見えるのだとしても，やはりより美しいリストを書いていたい。これは信念というより，常に新しいものを見てみたい，という欲の現れの一部なのだろう。だいいち，なにか新しいことができる，という期待がないと，とてもプログラムを書く気力が湧かないのだ。

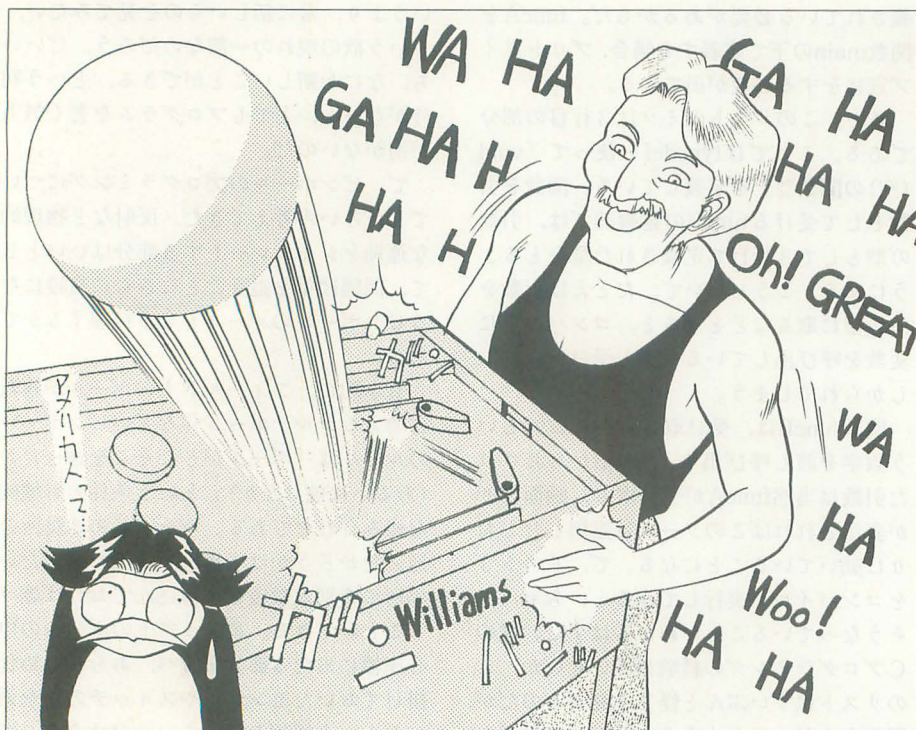
で，ピンボールのプログラミングについてもいろいろ考えてきた。反射など物理的な運動をシミュレートする部分はいいとして，問題は年を追うごとにさらに複雑になるピンボールのルールをどう料理するかである。

基本的に，フィールド上のボールが複数に増えるマルチボール化などのピンボールのルールは「ボールがどこそこを通ったか(あるいは触れたか)」という条件の有機的な組み合わせである。ボールがある場所を通ったかどうかは，裏マップ上の調べたい場所に特別な数値を埋め込んでおけば調べることができる。裏マップ上のボールの中心座標に当たる部分を調べ，あらかじめ仕掛けておいたセンサーやスイッチの上を通ったという情報を得ると，switch文などで処理を分岐させ，それぞれの処理に振り分ける。

分岐した先の関数では，ボールが通ったという情報をフラグかなにかに埋め込み，一定の条件が満たされていればマルチボール化などのルールにのっとったイベントを発生させる。これで一向に問題はないように見えるのだが，実はそうではない。

まず，画面を書き換えるタイミングが問題である。たとえば，ボールがあるスイッチを押すことによってすぐそばのランプが点灯する，というような仕掛けがあるとしよう。ボールがスイッチの上を踏んだ瞬間，そのスイッチの処理を司るサブルーチンが呼び出される。このときついでに画面の書き換えもやってしまいたいのだが，これでは不都合がある。ご存じのとおり，画面の書き換えはモニタの垂直帰線期間中に行わないと，画面にちらつきが起こるからだ。

リスト3

```
 1: #include "stdio.h"
 2:
 3: typedef void vfunc();
 4:
 5: void      funcA(int n)
 6: {
 7:           printf( "%d",n );
 8: }
 9:
10: void      main()
11: {
12:           funcB(funcA);
13: }
14:
15: void      funcB(vfunc f)
16: {
17:           f(99);
18: }
```

そこで画面の書き換えを行うかどうかを決めるためのフラグを用意し，帰線期間中にこのフラグを調べ，もし必要なら画面書き換えのルーチンを呼び出す。当然，ここでもボールの通った場所を記録する場面で使ったのとそっくりのswitch文をリストに埋め込むことになる。これではリストがスマートでなくなってしまう。

ほかにもある。ボールが通ったかどうかを記録するためのフラグというのは，当然グローバル変数として宣言される。画面書き換え用のフラグもそうだし，そのほか，とある場所をボールが通ってからの経過時間など，複雑なピンボールのルールをプログラムで実現しようとするとたくさんのグローバル変数を取ることになる。

そこそこの大きさをもつプログラムを組んでいると，このグローバル変数というのが意外に厄介者になるのだ。さきに挙げた関数名と変数名との衝突という問題は氷山の一角である。リスト中で使う識別子の総数が増えれば増えるほど，#defineの定義と衝突して気づかぬうちにとんでもないバグを埋め込んでいる，ということにもなりかねない。

このような問題点を避けるためには，関数の独立性を高めればいい。たとえば，関数内で使う変数領域は自前で確保するようにするなどといった手法を取るのである。ただし半面，関数の独立性を高めると，関数の呼び出しなどが繁雑になってしまう。そこで，効率よく独立関数を管理してくれるような仕組みが必要になってくる。

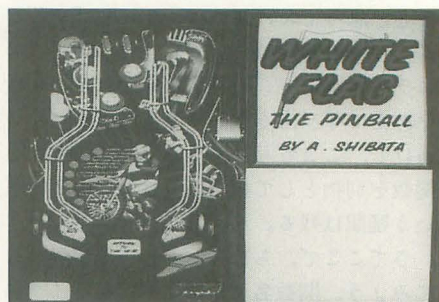
ここで思い出してほしいのが，関数のアドレスが変数に代入できる，ということ。要するに，関数のアドレスやパラメータ類を格納するデータ領域のポインタなどをまとめる構造体を定義してしまうのである。で，この構造体のメンバに「ボールがある場所を通ったら呼び出される関数のアドレス」とか，「ある仕掛けを描画する関数のアドレス」などを，プログラムの冒頭で代入しておく。あとは，裏マップを調べてswitch文で分岐させる代わりに，この構造体のメンバの関数を呼び出すようにする。

なお，この場合の利点は，

1.リストをしっかりと分割しておけば，飛び先の関数を書き換えたとき，コンパイルされるファイルが最小限ですむ

2.保守が比較的容易

などとなるだろうか。

C言語のプログラムというのはアセンブラに比べて静的なイメージがあると思うが，こうして見るとCでも工夫をすればけっこう柔軟性のあるプログラムが組めるものなのだ。このように呼び先を構造体のメンバとしてもっておけば，たとえばひとつのプログラムで複数のピンボール台を楽しめるようなものを作るときに重宝する。台によって仕掛けの種類が違うのだが，台を変えるごとに構造体のメンバを定義し直せば，プログラムはまったく違った動きを見せる。一方，処理をswitch文などで分岐させる方法を取ると，台ごとに違う分岐文をもたせ

制作途中の画面

なければならず，速度面からいっても前者のほうが断然有利である。

「要するにオブジェクト指向なんです」といったほうが，わかりやすい人もいるかもしれない。どうせやるならC++のようなそれ専用の処理系を使ったほうが手軽なのだろうが，X680x0でGCPPを使っている人というのを見かけたことがないので，とりあえずCでやっている。

ところで，肝心のピンボールはどうなっているかというと，先月号の付録ディスクに収録するべくそこそこのところまで作ったのだけれど，なんでも容量不足ということで載らないことになってしまい，制作は小休止。

制作中のプログラムでは，今回の「サブルーチンやデータ領域を構造体で管理する」という手法をもう一歩進めて，ごく簡単なイベント管理を行っている。つまり，ボールがセンサーなどのある場所を通った，といったときだけでなく，プログラムのどこかからの要求があれば，目的のルーチンが呼び出されるような仕掛けを作ってあるのだ。

遅延呼び出しのようなこともできるので，たとえば3秒後にとあるライトをつける，というイベントを描画イベント用のイベントキュー(呼び出されるイベントを保存しておく配列のようなもの)に登録すると，ちゃんと3秒間待ってからライトが点灯する。これと同じことを何も考えずにやろうとすると，時間のカウンタとかライトをつける動作が終了したかどうかのフラグだとか，これだけで数個のグローバル変数をとり，なおかつメインループごとに目的のライトの点灯を管理するルーチンを呼び出す，という無駄な処理が必要になるだろう。

ところでこの手法，応用次第でかなりすごいことができそうである。具体的には，「プリンス・オブ・ペルシャ」のようなアクションゲームを作るときなど，この柔軟性が威力を発揮するだろう。このあたりは，機会があればもう少し詳しく解説したいと思っている。

(つづく)

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1994年11月18日の到着分までとします。当選者の発表は1995年1月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞には当選できない場合がありますので，ご了承ください。

1

2名

X68000用　5″2HD版　12,800円(税別)

クレスト　☎03(3418)5993

先月号で紹介したカード型データベースです。バージョンアップにより，毛筆印字機能もつきました。

スターラスター

3名

X68000用

5″2HD版　5,900円(税別)

電波新聞社　☎03(3445)6111

ビデオゲームアンソロジーシリーズではありませんが，懐かしのゲームのX68000アレンジ版。オリジナルモードもあります。

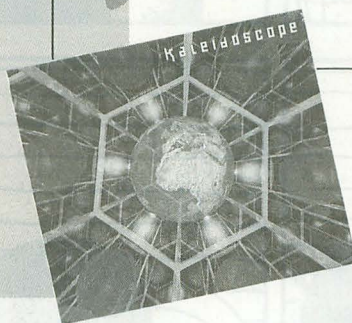
3 クイーン・オブ・デュエリスト外伝α＋

X68000用　5″2HD版　5,800円(税込)　5名

TAKERU　☎052(824)2493

女性キャラにこだわりたい人向けの対戦格闘ゲーム。漫画家や声優によるマニアックな一品です。

3名

MYU-RECORDINGS

読者の米山正晃さんから送られてきた同人CDをステッカーつきでプレゼント。高橋哲史氏の作品も収録されています。詳しくは今月号の「善バビ」を見てね。

CD用キャリングケース「PAPARA」

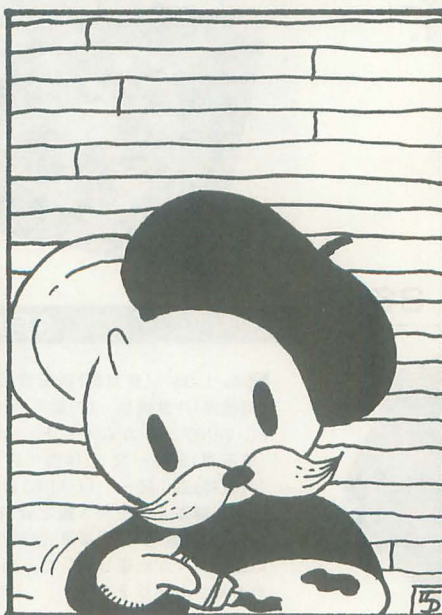
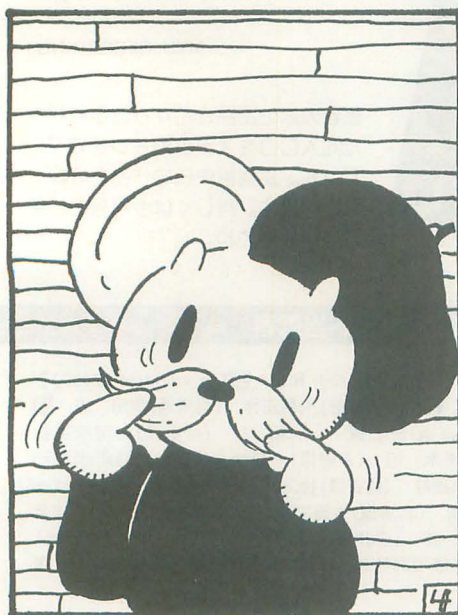
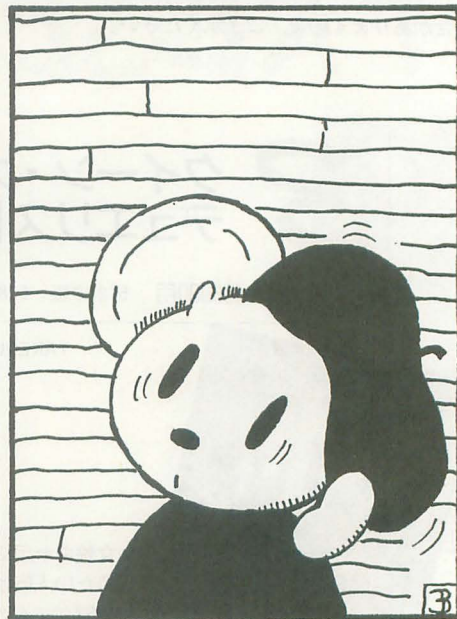
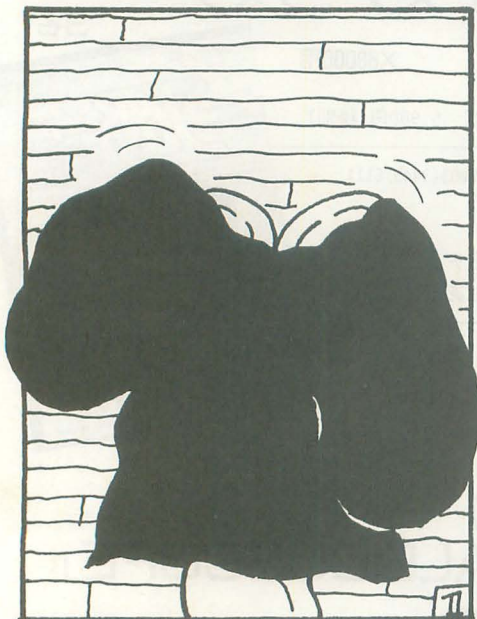
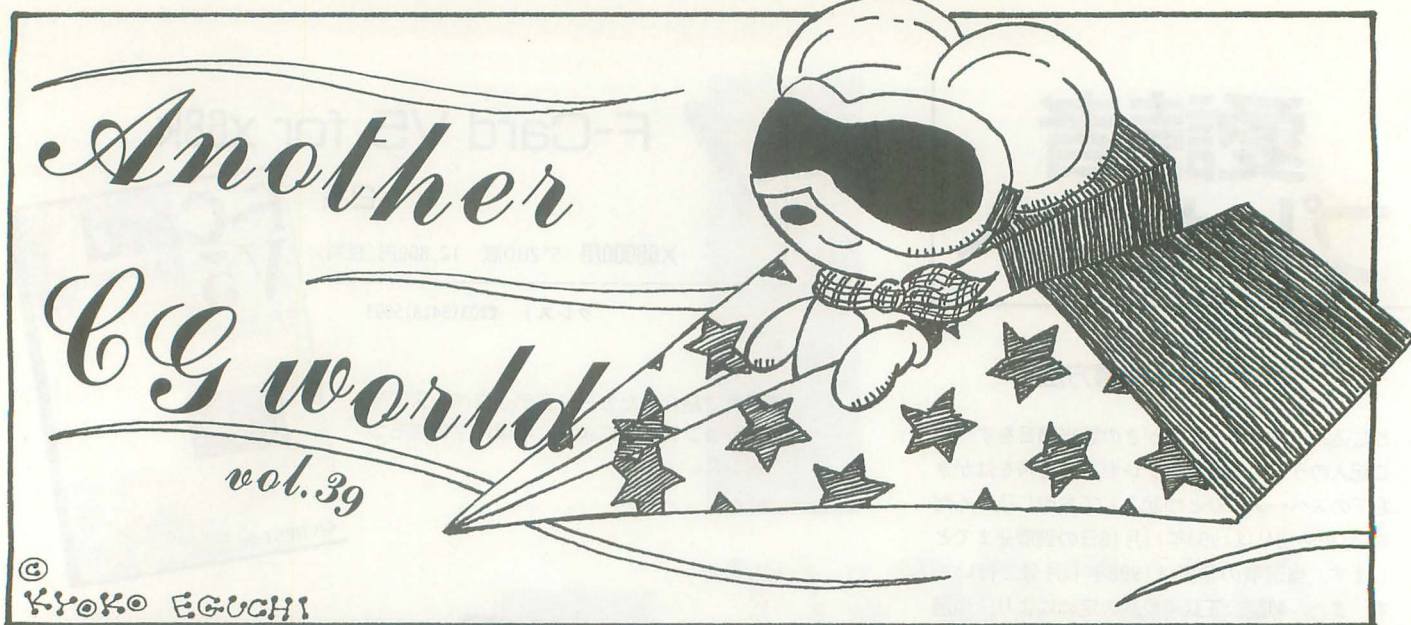
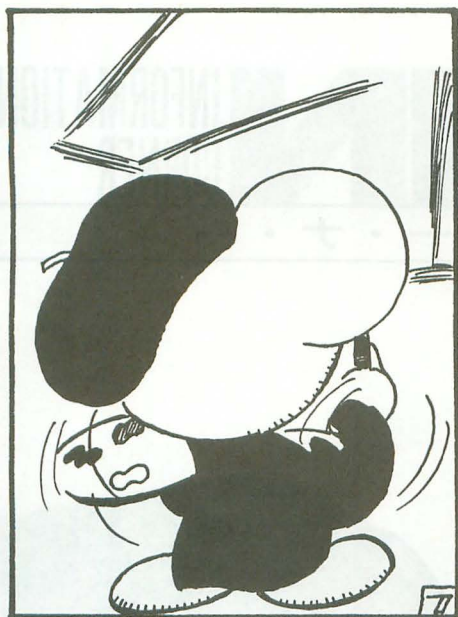
3名

横浜エレコム　☎045(475)2242
3,000円(税別)

カラフルなCD用キャリングケース。名前は「PAPARA」。CDが12枚収録できます。もちろんCD-ROMでもOK！　3名にプレゼントです。

9月号プレゼント当選者

1 Mu-I GS　(東京都)折田貴弘　(神奈川県)松井和宏　2 The World of X68000 II　(宮城県)片倉純也　(三重県)兼松隆文　(大阪府)永易正博　(兵庫県)青田　誠　3 PC-VANウェルカムキット　(北海道)三上誠一　加藤　幸　(福島県)大河原正樹　(埼玉県)武藤一文　(神奈川県)黒木　恒　(長野県)小松大介　(愛知県)岡本和己　(大阪府)森山隆一　(石川県)目黒雄峰　(佐賀県)村上淳一　ほか20名　4 スーパーリアル麻雀P IV原画＆設定資料集　(北海道)太田志輝　(青森県)本田知則　(神奈川県)岩崎敦司　(千葉県)伊藤耕一郎　(福井県)岩崎義介　(敬称略)
以上の方々が当選しました。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。

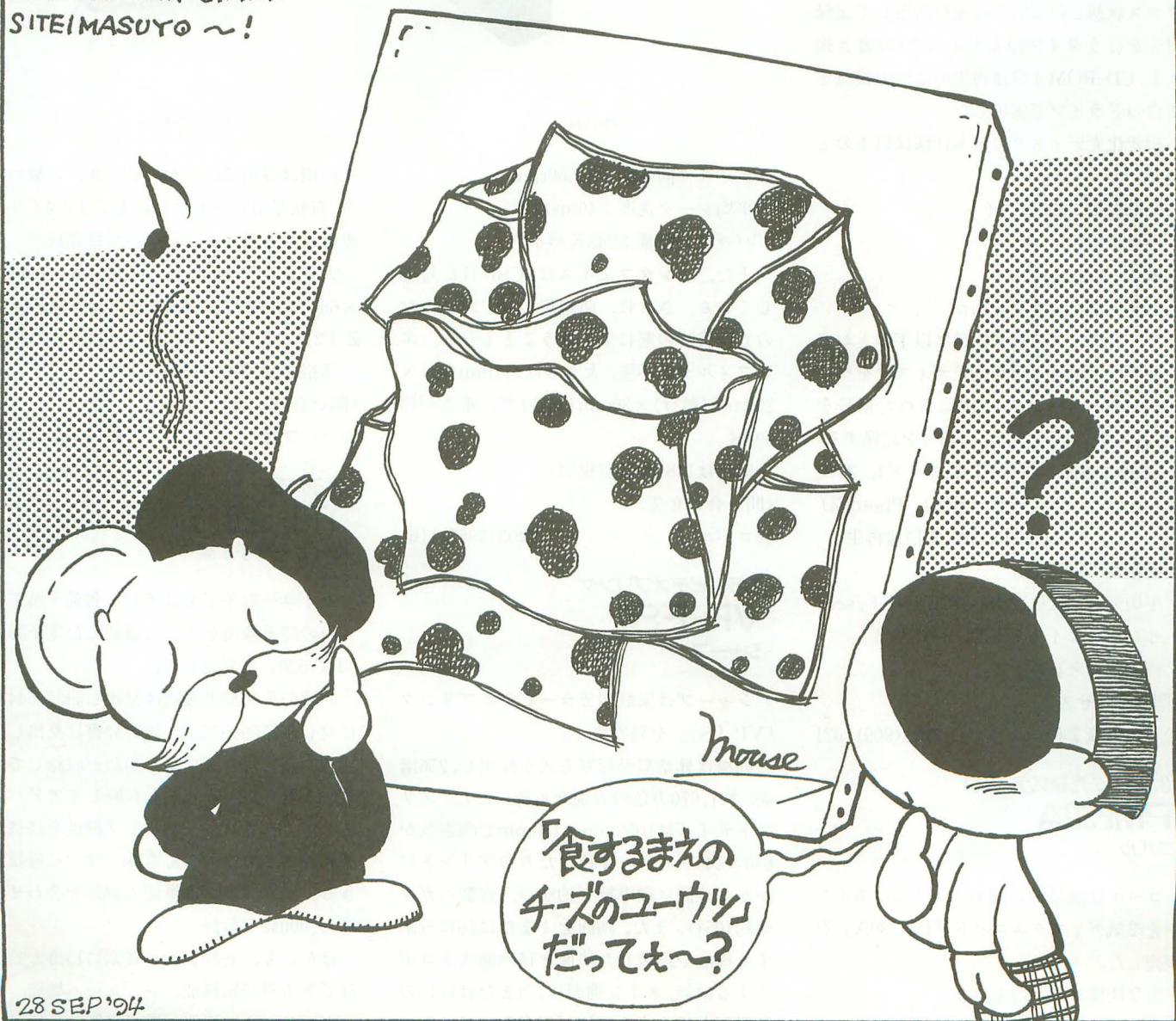
Another CG world
vol.39
© KYOKO EGUCHI

OKAMURA NAOYA SAN
ITSUMO TANOSIMINI
SITEIMASUYO ～!

mouse

「食するまえの
チーズのユーウツ」
だってぇ～?

28 SEP '94

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

光ディスクシステム
PD
松下電器産業

松下電器産業は新しいコンセプトの光ディスクシステム「PD」を開発した。

同システムは容量が650Mバイトの相変化光ディスク(記録膜の結晶状態とアモルファス状態での反射率の差を利用して記録再生を行うタイプの光ディスク)の書き換えと,CD-ROM 4倍速再生の2つの機能を1台のドライブで実現した。

相変化光ディスクの基本仕様は以下のとおり。

記憶容量：650Mバイト
外径：120mm
厚さ：1.2mm
トラックピッチ：1.2μm

また，ドライブの基本仕様は以下のとおり。

書き換え機能：相変化光ディスクをダイレクトオーバーライト(記録された信号を消去しながら同時に新しい信号を記録する)
CD再生機能：CD-ROM(モード1，2)
オーディオCD，PhotoCD
CD-ROMの4倍速再生
回転数：2026rpm
平均データ転送速度：870Kバイト/sec
インタフェイス：SCSI-II対応

商品化は来春の予定。

〈問い合わせ先〉
松下電器産業㈱　☎06(909)1021

3.5インチ光磁気ディスク
PM230A
コパル

コパルは携帯性に優れた小型の3.5インチ光磁気ディスクユニット「PM230A」を発売した。

主な仕様は以下のとおり。

記憶容量：128/230Mバイト

PM230A

ディスク回転数：3,600rpm
平均シーク速度：40ms以下
バッファ容量：243Kバイト

また，インタフェイスはSCSI-IIに対応している。さらに，防塵用のドアがついたので携帯時の塵に気を使うこともない。エアフィルタも不要。大きさは117mm(幅)×209mm(奥行)×30mm(高さ)で，重さが約695g。

価格は148,000円(税別)。

〈問い合わせ先〉
㈱コパル　☎03(3965)1161

カラービデオプリンタ
VP-ES1
シャープ

シャープは家庭用カラービデオプリンタ「VP-ES1」を発売する。

同機は昇華型熱転写方式を採用し,256階調,約1,670万色の表現を可能にした。プリントサイズは1052mm×100mmで画素数が640×448ドット。1枚当たりのプリントにかかる時間は標準紙で約80秒，官製ハガキが約105秒。また，画面を4または16に分割する機能や連続した映像を16画面ストロボにする機能,メイン画面に1/4または1/16の子画面を挿入する機能が用意されている。

VP-ES1

印刷は専用紙のほか，シール，官製ハガキ,布転写用シートなどにもできる(インクカセットとセットで，すべて別売)。

大きさは250mm(幅)×285.5mm(奥行)×66.5mm(高さ),重さが1.8kgと従来機(GZ-P21)に比べて小型化が実現されている。

価格は58,000円(税別)。

〈問い合わせ先〉
シャープ㈱　0120(078)178

電子辞書
TR-520
セイコー電子工業

セイコー電子工業は英和・和英・漢字の3つの辞書機能を1台に搭載した電子辞書「TR-520」を発売した。

同機の英和辞書機能は見出し語37,044語に対し訳語59,986語，和英辞書は見出し語18,203語に対し訳語30,109語を収録している。また，綴りの正誤を判断してアドバイスしてくれるスペルチェック機能を搭載し，英和辞書とも連動して意味もすぐに確認できる。漢字辞書には単語と熟語を合わせて約40,000語を収録。

ほかにも，各辞書別に任意に13語まで登録できる単語帳機能，レジューム機能，オートパワーオフ機能，電卓機能などが用意

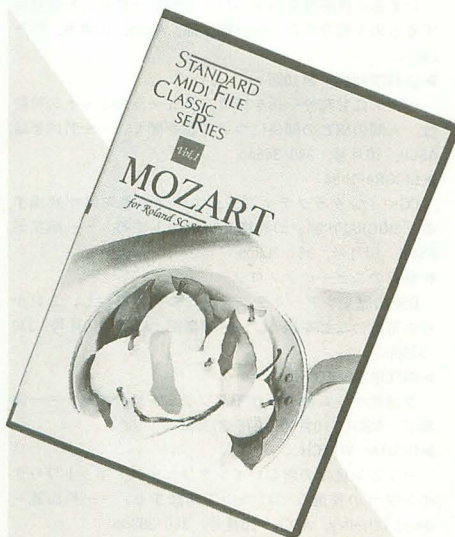
TR-520

されている。

価格は8,000円(税別)。

〈問い合わせ先〉

セイコー電子工業㈱　📞0120(052)440

SC-88対応音楽データ KT88シリーズ
サンワード

KT88-1001

サンワードはローランドの「SC-88」に対応した音楽データ「KT88-1001」ほか5本を発売した。

この音楽データはスタンダードMIDIファイルの形式で記述されている。タイトルは順に,

MOZART「モーツァルト序曲集」
TCHAIKOVSKY「くるみ割り人形」組曲
BIZET「カルメン」組曲より
J.STRAUSS「美しき青きドナウ」
MOZART「交響曲 第40番」
MOZART「交響曲 第41番」

である。ディスクのフォーマットは3.5インチ2DD(MS-DOS)と3.5インチ2HD(Macintosh)の2種類が用意されている。また,SC-55などの音源でも再生は可能だが,発音数の関係で音が抜ける場合がある。

X680x0で再生する場合,同社の「Mu-1GS」や,Z-MUSICとMZP(X680x0コレクションACE:秀和システムトレーディング刊などに収録)が必要になる。

価格は,それぞれ3,500円(税別)。

〈問い合わせ先〉

㈱サンワード　☎044(855)4335

INFORMATION

ヒューマンクリエイティブスクール・エンターテイメント・スピリッツ'94
ヒューマンクリエイティブスクール

ヒューマンクリエイティブスクールは昨年の第1回に続いて「ヒューマンクリエイティブスクール・エンターテイメント・スピリッツ'94」と銘打った学園祭を行う。

開催日は11月12日,13日の2日間。場所は渋谷BEAMホールとパルコパート1隣特設広場の2カ所となっている。

渋谷BEAMホールではゲーム関係の著名人によるライブや生徒が作成したゲームの大会,「ヒューマンコンピュータエンタテイメントコンテスト'94」の表彰式,トークショーなどが行われる予定。特設広場ではヒューマンの新作ゲーム大会など各種イベントが用意されている。

入場は無料。

〈問い合わせ先〉

ヒューマンクリエイティブスクール

☎0422(23)1111

SapporoMultimedia & CG '94
SapporoMultimedia&CG'94実行委員会

11月22日,23日の2日間にわたって「SapporoMultimedia&CG'94」が開催される。会場は札幌パークホテル。

同イベントは「見て,聞いて,触れてMultimedia」をテーマにCG作品のコンペティション,セミナー,Multimedia&CGアートショーなどが行われる。

入場はイベントにより有料のものと無料のものがある。

〈問い合わせ先〉

SapporoMultimedia&CG'94実行委員会

☎011(807)6122

NICOGRAPH '94
日本コンピュータ・グラフィックス協会/日本経済新聞社

11月15日~18日の4日間にわたって「NICOGRAPH '94」が東京・池袋「サンシャインシティ」と東京・大手町「日経ホール」の2会場に分けて開催される。

同展では「デジタルメディア」「インタラクティブ」「3D」「パーソナル」をキーワードにCG/CADの最先端テクノロジーを紹介する。主な内容は機器展示とコンファレンスで成り立つ。

機器展示は「サンシャインシティ・コンベンションセンターTOKYO」で行われ,入場料は一般が1,000円,学生が500円。開催時間はAM10:00~PM5:00となっている。

コンファレンスには講演やフィルムショーなどがあるが,日時,入場料はさまざまである。

〈問い合わせ先〉

コンファレンスに関して

㈳日本コンピュータ・グラフィックス協会

☎044(855)4335

機器展示に関して

日本経済新聞社事業局総合事業部

☎03(3243)9082

価格改定のお知らせ REDZONE
満開製作所

満開製作所は現在発売中のX68000CompactXVI改「REDZONE」の価格を130,000円から98,000円(ともに税別)に改定した。また,上記の金額に5,000円(税別)の上乗せで「REDZONE」の内蔵3.5インチドライブ2DD化改造ができる。

同機はX68000CompactXVIを改造したもので,10/16/24MHzの3モードを搭載している。保証に関しては従来どおり1年間,その後も有償での修理が受けられる。

〈問い合わせ先〉

㈱満開製作所　☎03(3554)3856

FILES

Oh!X

このインデックスは，タイトル，注記——著者名，誌名，月号，ページで構成されています。山々の樹に繁る葉もすっかり紅葉を迎えるなか，たまには移りゆく季節を感じながら芸術の秋でも満喫してみてはいかがでしょうか。

参考文献

I/O　工学社
ASAHIパソコン　朝日新聞社
ASCII　アスキー
コンプティーク　角川書店
C MAGAZINE　ソフトバンク
電撃王　主婦の友社
PIXEL　図形処理情報センター
マイコンBASIC Magazine　電波新聞社
My Computer Magazine　電波新聞社
LOGIN　アスキー

一般

▶NEWS
電通が電子ニュースペーパーの試作版を開発した話題など。——編集部，ASAHIパソコン，9・15号，8-9pp.

▶初心者がつまずく108の悩みを救う！
ハード，ソフトメーカー11社のユーザーサポートから，よくある質問とその回答を紹介する。——編集部，ASAHIパソコン，9・15号，16-27pp.

▶機械用言博物館 16
「終了する」をキーワードに，アプリケーションの終了について考える。——荻窪圭，ASAHIパソコン，9・15号，132-133pp.

▶GLOBAL INTERFACE
アメリカでのインターネットの実態を探る。——服部桂，ASAHIパソコン，9・15号，132-133pp.

▶ハードウェアFLASH！
ロジテックの4倍速CD-ROMドライブ「LCD-700」など，9月下旬までのハードウェア発売データを掲載。——編集部，LOGIN，18号，40-43pp.

▶今が旬だぜCD-ROM
CD-ROMソフト制作現場の取材やライターお勧めのCD-ROMソフトの紹介から，CD-ROMの可能性を探る。——編集部，LOGIN，18号，115-131pp.

▶カーナビゲーションシステムのススメ
ナビゲーションシステムの解説と，車や自転車，気球に搭載したレポート。——編集部，LOGIN，18号，156-163pp.

▶くねくね科学探検　第5回
NTT基礎研究所の内藤誠一郎氏を迎え，目の錯覚について考えていく。——鹿野司，LOGIN，18号，182-185pp.

▶ゲームミュージック・フェスティバル'94
7月30，31日に日本青年館で行われたゲームミュージック・フェスティバルの模様をレポート。——宮田まこ，コンプティーク，10月号，108-109pp.

▶NEWS COLLECTORS
3DOにPowerPC搭載64ビットマシンアダプタ登場の噂など。——編集部，電撃王，10月号，20-23pp.

▶特集2　次世代機の10大秘密
次世代ゲーム機をめぐる動きの最新情報。関係者による次世代機クロスレビューなど。——編集部，電撃王，10月号，46-55pp.

▶シリーズ謎と真実
通信カラオケをテーマに，その仕組みを解説。各メーカーの方式の特徴を明らかにする。アニメ，特撮の主題歌のラインナップ分析つき。——編集部，電撃王，10月号，162-165pp.

▶FAXモデムで情報達人になろう！
FAXモデムの解説から始まって，商用ネットやメーカーサポートの紹介，チケット予約を実際に行うなどモデムを活用する。——編集部，マイコンBASIC Magazine，10月号，35-45pp.

▶CD-ROMからはじめるマルチメディア　第6回
オリジナルCD-ROMを作ってもらえる店「CD倶楽部」を紹介。CD-ROMドライブの新製品情報あり。——吉岡哲也，マイコンBASIC Magazine，10月号，46-49pp.

▶新ハード特捜部
「SEGA SATURN」の本体カラー変更のニュースや新作の画面写真を一挙に公開。——山下章，マイコンBASIC Magazine，10月号，147-151pp.

▶Arcade Game Graffiti　第8回
1981年のアーケードゲームを振り返る。ニチブツの「フリスキートム」やコナミの「フロッガー」，任天堂の「ドンキーコング」が登場。——編集部，マイコンBASIC Magazine，10月号，158-161pp.

▶NEWS & VIEWS
ジェット旅客機B777に息づくコンピュータシステムの秘密に迫る。——鍛治信太郎，ASAHIパソコン，10・1号，12-13pp.

▶TEST RESULTS
ペンで入力して無線通信ができる情報機器「MESSAGE TC301」などを紹介する。——編集部，ASAHIパソコン，10・1号，28-37pp.

▶機械用言博物館　17
今回は「アップグレードする」をキーワードにメーカーのサービスについて考える。——荻窪圭，ASAHIパソコン，10・1号，132-133pp.

▶特集　ゲームエンディング解体新書
ゲームのエンディングについてソフトハウスの姿勢を聞くほか，各ゲームのエンディングまでのポイントを授ける。——編集部，LOGIN，19号，123-137pp.

▶人工生命M1号
Artificial Lifeの現状を紹介し，その将来について日米の研究者にインタビューする。——編集部，LOGIN，19号，160-163pp.

▶ネットワーク新時代
新しい時代の通信の動きについて紹介する。14400bpsのモデム，GUIによる通信，通信サービスなど。——編集部，LOGIN，19号，172-175pp.

▶架想楽園へ行こうVer.2.03 No.8
純粋デジタルのキャラクターたちを生み出す工房「The Works」のレポート。——中田宏之，LOGIN，19号，186-189pp.

▶くねくね科学探検　第6回
芝浦工業大学の武田邦彦教授を迎え，化学とコンピュータについて考える。——鹿野司，LOGIN,18号，182-185pp.

▶特集　電脳エンターテイメント
次世代ゲーム機など，コンピュータを使った娯楽の現状と将来像について考える。——長谷川博之ほか，I/O，10月号，35-58pp.

▶特集"高速"通信環境
FAXモデムに関する解説と最新FAXモデムの製品ガイド。——編集部，I/O，10月号，114-121pp.

▶Multi Media Watching 10
アメリカの衛星デジタル放送のニュースやインターネットなどを取り上げる。——奥野雅之，I/O，10月号，154-157pp.

▶携帯情報端末でGO OUT！
いまある携帯情報端末のハードやサービス，今後登場するものを紹介する。——編集部，ASCII，10月号，257-280pp.

▶新科学対話　第10回
ゲストに甘利俊一氏を招き，ニューラルネットの可能性，人間の脳との関係について話を聞く。——竹内郁雄，ASCII，10月号，340-346pp.

▶SIGGRAPH'94
CGやインタラクティブテクニックの最先端が登場する「SIGGRAPH'94」の模様をレポートする。——編集部，ASCII，10月号，361-368pp.

▶魅惑のニューテクノロジー
DSPの歴史やアーキテクチャについて解説し，これからの可能性などを探る。——編集部，ASCII，10月号，370-375pp.

▶INTERCOOLED
次世代ゲーム機の新作予定ソフトを紹介する。——編集部，ASCII，10月号，376-379pp.

▶DIGITAL WATCH
パソコン通信の新しいインタフェイス，ネットワークインフラの整備などについて対談する。——桝山寛＋David D'Helley，ASCII，10月号，380-383pp.

▶稀代もののけ考
ハイテクグッズ大集合のページ。今回はちょっと変わったテクノロジー搭載のグッズを集める。——バカパパ，ASCII，10月号，444-445pp.

▶光磁気ディスクの勧め　その3
光磁気ディスクをバックアップメディアとして使う場合のコツを伝授する。——佐田守弘，My Computer Magazine，10月号，55-59pp.

▶ザウルスをパソコンと併用するためのソフト
ZAURUSの機能をパソコン上でエミュレーションする「Hyper HAL-CATCH Ver2.0」を紹介する。——編集部，My Computer Magazine，10月号，84-87pp.

▶大学におけるマルチメディア・ネットワークとCG/CAD教育の現状（CG/CAD教育編）
各大学でのCG/CAD教育の現状とこれから求められる方向性について，取材とアンケートから探る。——編集

部，PIXEL，10月号，89-124pp.
▶特集　低価格で高品質化進むカラープリンタ
カラープリンタの現状と各社の製品52機種を紹介する。──編集部，PIXEL，10月号，125-137pp.

X1 turbo/Z

X1シリーズ
▶Magical Crystal
同じ色のクリスタルを4つ並べて消すパズルゲーム。──電脳SOFT，マイコンBASIC Magazine，10月号，111-113pp.
▶スーパーストリートファイターII～キャミィ～
NEW-FM音源ドライバ用のミュージックプログラム。──川村賢治，マイコンBASIC Magazine，10月号，124-125pp.

X68000

▶未確認クリエイターズ
プログラム部門の入選作品に，X68000用バラエティゲーム「ガーディアンズ」が選ばれた。──編集部，LOGIN，18号，150p.
▶S.S.express
パソコン通信でゲームが買える「ASCII-net SoftShop」を紹介。X68000用の新作は「スーパーストリートファイターII」など。──編集部，コンプティーク，10月号，39-58pp.
▶カルトG
今回は同人ソフトについて触れる。X68000用「過労伝説」がちょっと紹介されている。──平野裕子，コンプティーク，10月号，130-131pp.
▶新作王
X68000用は電波新聞社の「スターラスター」とカプコンの「スーパーストリートファイターII」。──編集部，電撃王，10月号，170-171pp.
▶電撃新作予定表
発売が予定されるソフトの一覧表。X68000版「プリンセスメーカー」「上海IV」など。──編集部，電撃王，10月号，188p.
▶頭脳トレーニング
3種類のミニゲーム集。ストップウォッチゲーム，数字暗記ゲームなど。──萬道賢治，マイコンBASIC Magazine，10月号，114-115pp.
▶FULLMETAL FIGHTER
対戦型ボクシングゲーム。──少年一号，マイコンBASIC Magazine，10月号，116-118pp.
▶真・女神転生2
ミュージックプログラム。──上古仁志，マイコンBASIC Magazine，10月号，126-129pp.
▶スーパーストリートファイターII
X68000用「スーパーストリートファイターII」のプレイレポート。──板場利光，マイコンBASIC Magazine，10月号，167p.
▶SUPER SOFT HOT INFORMATION
X68000用は「スターラスター」や「スーパーストリートファイターII」など。──編集部，マイコンBASIC Magazine，10月号，付録13p.
▶NEWSOFT
新作ソフトの紹介。X68000用には「スターラスター」。──編集部，LOGIN，19号，22p.
▶AV STRASSE
X68000用「スーパーストリートファイターII」をコラムで紹介。──編集部，ASCII，10月号，393-396pp.
▶ONLINE SOFTWARE INDEX
大手ネットにアップロードされたソフトを紹介する。X68000用CD→PCM変換ソフト「cd2pcm.x」など。──編集部，ASCII，10月号，491p.
▶パソコン研究室
歴代のPC-9801のBASICによる実行速度比較。参考資料としてX68000(10MHz)のデータを掲載。──Space Club，My Computer Magazine，10月号，115-117pp.
▶なんでもQ＆A
「SX-WINDOW ver.3.1」の「シャーペン.X」にある“描画モード”についてとHuman68kのバッチファイルに関しての質問に答える。──シャープ，My Computer Magazine，10月号，148-149pp.
▶SX-WINDOWプログラミング　第12回
シャーペンの外部コマンド作成ライブラリについての2回目。前回に作成したライブラリの問題点とその解決，簡単な外部コマンドの実例を扱う。──吉野智興，C MAGAZINE，創刊5周年記念号，123-127pp.
▶Information from Compiler Makers
「Workroom SX-68K」と「SX-WINDOW ver.3.1」を同時に使う場合の留意点および便利な使い方を紹介する。──シャープXグループ，C MAGAZINE，創刊5周年記念号，156-163pp.
▶創刊5周年記念特別CD-ROM
創刊号から特別付録に収録されたプログラムをCD-ROMに再収録。──編集部，C MAGAZINE，創刊5周年記念号，166p.

ポケコン

PC-E500
▶バッタと水溜り
反射神経養成ゲーム。──はたらいたはらたいら，マイコンBASIC Magazine，10月号，119p.

新刊書案内

技術立国日本が危ない！
宝島社刊
☎03(3234)4621
A5判　263ページ
1,100円（税込）

バブルがはじけて以来，日本の将来を危機的な視点で語る本は多く出ているし，そう語るのが流行のようになってきた。多くは経済システムの話で，流通がどうしたこうしたとか，官民の癒着がどうしたこうしたという，昔からの視点にとらわれたものが多い。そんななか，「ものづくり」の視点から危機を唱えるという本が登場した。しかも，別冊宝島シリーズだ。特に日本を儲けさせ，地位を向上させたのは，半導体と自動車だ。この2つを抜きにして日本経済の発展は語れなかったはずだ。本書はその2つを重点的に突く。
まず「半導体」方面。TRONの坂村健氏や東北大学の西澤潤一氏へのしっかりした取材記事があり，日本のベンチャー企業の実態（大手企業がベンチャーの開発した技術をパクっていることなど）や，ソフトの違法コピーに関するライターによる記事がある。2本の取材記事が本書全体の信憑性を増しているといっていい。結果としてOSからユニコード，理系学生の創造性から企業の現場，情報スーパーハイウェイやHD-TVに対する日本の反応などなどをいくつもの記事でうまくまとめている。
「自動車」のほうは，ボルボとトヨタの比較（どちらがいいという話ではなく），紙と鉛筆からCADへと設計手法が変わったことによる悪弊が出てきたこと，マツダがなぜ失敗したか，といった自動車業界の危機が語られる。「半導体」方面に比べると弱い気がするが，それはまだ自動車産業は頑張っているということかもしれない。
そして最後はノンジャンルで攻める。高速増殖炉に対する警告，日本の工業界を支えてきた中小の町工場がもっていた技術力とその衰退，H-IIロケット（そういえば，失敗したようだが）の問題点などなどが並ぶ。ここがいちばん圧巻かもしれない。技術や製造業の視点から日本を語る本としてかなりまとまっていると思うゆえ，一読を勧める。（K）

ENIAC神話の崩れた日
ATANASOFF
Forgotten Father of the Computer

ENIAC神話の崩れた日
クラーク・R・モレンホフ著
最相　力・松本泰男　共訳
工業調査会刊
☎03(3817)4701
菊判　270ページ
2,350円（税込）

世界で最初に作られたコンピュータ（電子計算機）といえば，多くの人が「ENIAC」のことを思い出すだろう。事実は違った。アイオワ州立大学教授だったアタナソフ氏がベリー氏とともに「アタナソフ・ベリー・コンピュータ」を開発した。ENIAC完成を遡ること数年前である。
本書はENIACの開発者モークリー氏の特許が無効であるとした裁判の資料をもとにしたノンフィクションである。これで，アタナソフ氏は世界で初めて電子計算機を発明したという名誉を回復した。しかし，この判決は1973年に出ていながら，広く知られていないことが残念である。

ブーメランはなぜ戻ってくるのか
西山　豊著
ネスコ刊
文藝春秋発売
☎03(3265)1211
四六判　237ページ
1,500円（税込）

ブーメランがどんなものか知らない人はいないだろう。
本書は誰もが名前は知っているけれども，実際に使ってみたり，「なぜ戻ってくるのか」という理由について知っている人がほとんどいないブーメランに焦点を当てている。内容はブーメランの歴史や科学的な分析などである。これが平易な言葉を用いわかりやすい。それ以外にも，ブーメランがなぜ戻ってくるのかを自分なりに考察し，ブーメランを自作してしまうところは興味深い。
著者の日常のふとした疑問について考えていく姿勢に学ぶべきところは多い。

QUESTION and ANSWER

1994.8月号の「内蔵AD PCM高音質化」改造をするために,「'94最新74シリーズIC規格表」(CQ出版社)と「絵ときディジタルICカラー版」(浅川　毅著,オーム社)を購入して,自分なりに勉強してみました。「74HC157の15番ピン(STROBE)をグランドに落とす」というのはこのピンをLレベルにするということなのでしょうか?　また,8月号に掲載されている実装写真では,74HC157の15番ピンが基板のいちばん上の左から,2番目の小さな1番ピンにつながっているように見えるのですが,どうなっているのでしょうか?(私はいちばん左上のICの8番につけましたが)

あと,PCMの音質が上がったのはよいのですが,本体(X68000XVI,20MHz改)の処理速度が落ちたような気がします。改造された方のなかに,このような症状が出た方はいらっしゃいませんか?

京都府　村上　浩二

村上さんの質問は3つあるのでひとつずつ,順番に答えていきましょう。

まず,ひとつめの「グランドに落とす」というのは,村上さんが考えているとおりL(ロー)レベルにするということです。たいていのデジタルICはLレベルがGNDです。

2つめの実装写真ですが,なるほどそう見えますね。基本的にGNDレベルに落とすということは,74系ICの右下のピンにつなげるのなら,村上さんが考えているとおりで間違いではありません。ただ,私がなぜそこで1番ピンにつけたか(過去の私は他人なので),忘れてしまって覚えていないのですが,きっとなにか意味があったんでしょう。残念ながら現在手もとに実機がないので細かな確認をすることができませんが……。

3つめの処理速度に関してですが,これは実は当たり前のことなのです。まず,X68000では通常,AD PCMの再生はDMAの3チャンネルを使って行っています。

さて,DMAで主に使われるモードには,サイクルスチールモードとバーストモードの2種類があります。普通,DMAを使うときはDMAが動いているあいだは,CPUはバスを開放しなくてはなりません(利用できない)。したがって,DMA利用中はCPUはバスアクセスできないので(メモリさえもアクセスできない),DMAが動いているときはCPUはほとんど停止しています(最近のCPUはちょっと違いますが)。

ただし,これはDMAのバーストモード動作の場合で,AD PCMで利用しているようなサイクルスチールモードの場合は,DMAはほんの少しずつバスを奪うので,見かけ上はDMAとCPUが同時に動いているように見えるのです。もちろん,DMA使用中はCPUのパワーはその分奪われています。

もうわかりましたね。

AD PCMが高音質化して,秒間の転送ビットが上がるということは,その分バスが占有されることが多くなるということです。すなわち,CPUがバスを自由に使える時間が減ってきて,処理が重くなってしまうというカラクリがあったのです。単に,データが倍の大きさになるから……と思っておけばいいのではないでしょうか。

(瀧　康史)

C言語でラスタースクロールするプログラムを作成したのですが,FM音源(OPMDRV3.X)を鳴らすと画面が乱れてしまいます。なにかいい解決法はないでしょうか。

岡山県　幡上　幸史

要するに割り込みの優先順位が問題になっているわけですね。

通常,音楽ドライバのような時間情報に大きく依存するドライバでは,データ出力の精度を確保するために割り込みを禁止することがあります。しかし,ラスタースクロールのように時間的にさらにクリティカルな処理を行う際にはそれを最優先させなければなりません。それには音楽ドライバ側に多重割り込みのためのモードが必要になってきます。

しかしOPMDRV3ではそのような設定が不可能ですから,ここではやはりZ-MUSICを使うことをおすすめします。Z-MUSICはゲーム用途にも使用できるように設計されていますから,うまく対処すればラスタースクロールなども同時に実現できるようになります。

そのほか,Z-MUSICでのゲームBGMや効果音をつける際に必要な情報はZ-MUSIC ver.2.0のマニュアル第14章をお読みください。

3月号の付録ディスクについてきたSX-WINDOWのカラーアイコンでは,クリーナなどのシステムのアイコンまでカラー化されていましたが,これはどのようにして設定したものなのでしょうか。変更方法を教えてください。

島根県　吉村　幹夫

えーと,質問の主旨がわからない人もいると思うのでざっと解説します。

SX-WINDOWのアイコンメンテでは,あらかじめアイコンを選択してから立ち上げれば,そのアイコンをエディットできます。しかし,ここで指定できるのはファイル1個につきアイコン1個です。つまり,クリーナーやディレクトリアイコンのように開いたり膨らんだりと,複数個のアイコンを使ってアニメーションするものについては,基本パターンしかいじれないわけです。

では,付録ディスクにつけたものはどのようにして描き換えたかというと,実は以下のような方法があるのです。

まずアイコンリストを開きます。アイコンメンテからリスト表示でもかまいませんし,アイコンリスト.Xを直接立ち上げてもかまいません。

そこで右クリックしてください。

メニューからアイコン表示を選択します。すると,普段はリストに出てこないものも表示されていることがわかります。クリーナなども見えるはずです。ここを書き換えてやればよいわけです。

ちなみにこれでも完全に書き換えられないのがアクティブウィンドウのレイヤーを変更するアイコンです。1枚だけなら書き換えられるのですが,操作をするとシステムが元のものに描き直してしまいます。こうなると,SXシェルを書き換えるしかないかも……。

私のディスプレイはCU21CDです。SAdjust.rでは9月号の10番の設定がOKでした。11番は直接指定はだめで,10番から段階を追って設定するとOKでした。SAdjust.rで段階を追って設定する(要するに10番を設定したあと11番を設定する)にはどうしたらいいでしょうか。スタートアップメンテに2つSAdjust.rを登録してもだめでした。

東京都　松嶋　竜

2つ登録するというのはコピーしたSAdjust.rの名前を変えて登録したのでしょうか？ 10月号の記事中で解説してあったと思いますが，Adjust系のツールを複数個登録することは絶対に避けてください。実画面モード時のスクロール処理がおかしくなります。万一，表示がおかしくなったまま終了してしまったときは，ESYSD.X(これも付録ディスクに収録されている）で余分なAdjustを削除してください。

本題に入りましょう。

質問にあるCRTCレジスタの設定を2段階で行わなければ設定されないというのは，もしかしたらSAdjust.r側の不備かもしれません。現在，状況によってはパラメータがうまく設定されないことがあることがわかっています。普通に使うには特に問題のない範囲だとは思いますが……。

さて，CRTCのレジスタ設定などを2重に行う方法ですが，全自動での設定をやりたいのでしょうか？

作業を手動で行うならデスクトップ用のメニューに10番と11番の設定を書いたSAdjust.rの起動メニューを追加すればよいでしょう。これでも実用上問題はないと思います。ウィンドウ上で実行する処理内容によって画面モードを切り換えるようにしておくのも便利ですから，いずれにせよそういった設定を加えておくことをおすすめします。

自動設定の場合ですが，まず「段階を追って設定すればOK」というのが，どの程度「段階を追って」なのかわかりませんので，ここでは2とおりの方法を紹介しておきます。

まず，スタートアップメンテでSAdjust.rを登録する際に，CRTCレジスタに設定する値，すなわち-Cの設定項目を二重にしてみてください。つまり-Cで10番の設定をしたあとに，もう一度-Cの設定を書いて11番の設定を書いておくわけです。コマンドラインは順次実行されますから，こうしておくと順番に設定されたのと同じことになります。

また，パラメータ設定後に多少時間を置かなければならないような場合も考えられます（急に解像度を変えるのはディスプレイに悪そうな気もしますし)。本当に臨界に近い部分ではディスプレイが「少し温まって」こないと有効でないとかいう状況もあるかもしれません。

こういったときにはスタートアップメンテで10番の設定を行っておいたうえで，SX-BASICを使ってSAdjust.rに，11番の設定内容に相当するメッセージを送ってください。

具体的なプログラムにすれば，

```
int n
n=findtskn("SAdjust.r",0)
sendmess(n,"CRTC ～")
```

のようになります。

しかし，臨界点でのディスプレイの使用はディスプレイに負担をかけますし，トラブルの原因になる可能性もありますので，できれば少しくらい余裕がある設定のほうがよいでしょう。快適なウィンドウ環境を追求するのもよいのですが，ディスプレイはあんまりいぢめすぎないようにしてくださいね。

SX-WINDOWを使っています。最近ハードディスクを買い換えたのですが，ファイルをつかんでほかのウィンドウに放り込むと元のファイルが消えてしまうようになりました。以前はそんなことはなかったと思ったので調べてみると，同じドライブ内だとムーブで，別のドライブだとコピーになることはわかりました。今度のハードディスクは1パーティションしか作っていないので，コピーしたいときにはわざわざ一度RAMディスクに持っていってコピーしています。そもそも，なんでこんな仕様にしているのでしょうか。仮想ドライブなどを使えばなんとかなるものなのでしょうか。

石川県　宮島　武司

おっしゃるとおり，SX-WINDOWでは同じドライブ内ではムーブ，違うドライブへはコピー動作になります。

マニュアルをよーく読んでいればわかることですが，SX-WINDOWを使う人はあまりマニュアルを読まないみたいなので(読まなくてもちゃんと使えるし……)，解説しておきましょう。CTRLキーを押しながら操作することにより，強制コピーにすることができます（さすがに同じディレクトリ内では無理ですが)。

そのほかにも，OPT.1キーを押しながらアイコンをダブルクリックをすると行入力モードでコマンド起動されますし（結構便利)，シフトキーを押しながら操作すると選択ファイルのファイルの追加指定になります。

ディレクトリウィンドウ内での右クリックメニューにシフトキーを併用できることを知らない人もいるようです（サイズ順，日付順などの表示指定ができる)。いろいろ触ってみればだいたいの勘所はつかめてくるでしょう。

ついでですから，シャーペンまわりのことも紹介しておきます。通常は変換キーのアサインがASK2とは違っているASK3もシャーペン (ver.3.0) 上で使用しているときに限りASK2互換のキーアサインになることはよく知られていますよね。それ以外にもいくつかの拡張が行われます。

日本語変換中に誤って確定してしまうことはよくありますが，こういうときはUNDOキーで変換前の状態に戻せます。似たような機能でシフト＋ESCを押すと，直前に変換した文字列を呼び出すことができます。また，XF1キーを押しながらキー入力するといわゆる透過モード（半角英数字がそのまま入力される）になります。

そのほか，シャーペンのデフォルトのキーマップ割り当てでは，XF2キーにOPT.1キーと同じ内容が割り当てられていることも覚えておくとよいかもしれません。

（中野　修一）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので電話番号も明記してください。

宛先：〒103　東京都中央区日本橋浜町3-42-3
ソフトバンク株式会社出版部
Oh!X編集部「Oh!X質問箱」係

FROM READERS TO THE EDITOR

秋もずいぶんと深まり，食べるものが美味しい今日この頃。あれを食べたりこれを食べたりと油断をしているとあっというまに……。そんなときには外に出て，ひと汗かいてみませんか？　新しい出会いがあるかもしれませんよ。

◆「メガディスプレイ追記編」を読んだあと，CRTC.Xを起動し，実用的な最大ドット数を求めるためにかなりの時間を費やしました。私のディスプレイはCZ-602Dですが，ノンインタでは816×576が画面をはみ出さないで使える限界（?）のようです。ちらつきを気にしなければ，もう少し広がるのですが……。
中嶋　康弘(35)群馬県

自分に合った環境を作ってX68000をしっかり使ってあげてくださいね。

◆ついに，230Mバイト，3.5インチMOドライブを購入しました。エレコムの「EMO-L230」です。X68000に接続して使用できるかどうか不安があったものの，MOドライブ裏側のディップスイッチ1，2を両方OFFに設定すればハードディスク（HDモード）として使用できました。ハード構成はX68000EXPERT＋シャープ純正SCSIボード＋「EMO-L230」です。ソフトは「Human68k ver.2.0」標準SCSIドライバだけで，MOドライブのマウント(接続認識)，フォーマット，読み書きができました。もちろん，128/230Mバイト両モード共に使用できます。大満足です。
渡部　裕亨(27)佐賀県

よかったですね。「おそらく大丈夫」と思っても，実際につないでみないと安心できないものですから。

◆友人の1Gバイトハードディスクを見ると自分の120Mバイトの狭さに悲しくなってくる。散乱した数百枚のフロッピー。圧縮されるだけで展開されないデータたち。すべては金欠病がいけないんだ（涙）。
角谷　光憲(20)愛知県

ハードディスクにデータを入れても今度はそれを展開している時間が無駄になってしまうので，いまのほうがかえっていいかもしれませんよ。

◆「デイトナUSA」のコース脇に配置されている樹木のポリゴン数はたったの8。平面を十字に交わらせ樹木のテクスチャを張り付けてあるだけです。善司さんのレビューにあったビデオで，コマ送りをしているときに見つけました。すごい性能をもったボードを載せても，ポリゴン数を減らすための努力やさらに見劣りさせない見せ方，どれをとっても感心せずにはいられませんでした。
鈴木　政宏(21)宮城県

できるかぎり無駄はなくす。やはり，贅沢は敵なんですね。ちょっと違うかな。

◆Oh!X（MZ）の作ってきた文化は「ないなら作っちゃえ！」の精神論と方法論だと思います。これこそが，パーソナルコンピュータの使い方のあるべき姿の一面であり，失われてはならない部分だと思います。これからもがんばってください……いや，ともにがんばりましょう！
松尾　和浩(33)新潟県

こういう皆さんがいるかぎり，失われることはないと思います。私たちもがんばらないといけませんね。

◆1Gバイトのハードディスクを使うようになって，いろいろ便利になりましたが，困ったことがひとつ。どこになにを入れたか忘れてしまっているのです。この間，久しぶりにハードディスクの中身を整理していたら，まったく同じフォントデータが別々のディレクトリにあったり，新しいバージョンをインストールしたはずのツールの旧バージョンがまだ残っていたりとほとんど無駄遣いをしています。トータルで200Mバイト程度しか使っていないので，探せばなんとか見つかりますが，これで1Gバイトがフルに埋まったら，どうなるか心配です。
田高　浩一(28)東京都

まだ空きがあるなら大丈夫。まずいと思えば自分でなんとかするもんです。そのときには遅いかもしれないけど……。

◆「スターラスター」が出る！　嬉しさのあまり，パソコン屋さんから「スターラスター」の広告を「がばっ」ともってきて，部屋中に張りまくってしまいそうです。わくわくわく。
林　大助(18)神奈川県

いまごろは部屋中に張りまくられたポスターに囲まれ，ハマリの状態でしょうか。

◆冷却用ファンのついていないパソコンは静かでいいなあ……。いつからだろう，ファンがつくようになったのは？
西井　貴(29)三重県

石のほうも人間の無理な要求に応えるためにキレる寸前まできているので少しは冷やしてやらないといけないのかも。

◆お盆休みで久しぶりに実家へ帰った。そして家にあるX68000PROIIを起動しようと思ったらぜんぜん反応がない。それもそのはず，なんと0ドライブが死んでいた。ショック！　4年間も元気に活動していたのに。悲しい。早く元気になってくれ～！
露崎　秀明(22)千葉県

あまりにも長い間放って置かれたので，マシンがすねてしまったのでしょうか？

◆値段が高くなっても買い続けるからネ。大きくなぁれ，大きくなぁれ，大きくなってアリさんになぁれ！
多田　智(19)高知県

いまはアリさんよりも小さいんですね（涙）。

◆日本橋で自分が下取りに出したX68000SUPER-HDとモニタが並んで売りに出されているのを偶然見つけてびっくり。「この子たち，うちの子なんです」とひとりで騒ぎ，連れに「恥ずかしいからやめなさい」といわれてしまった。
小林　紀子(26)大阪府

もちろんその子たちは引き取って帰ったんですよね。え，そのままですって？　人でなし（冗談）。

◀岡村　直也　兵庫県
これからの話は恵子ちゃんを主人公にして進むようです。これを見ると恵子ちゃんもなかなかやりますね。ちょっと詰めが甘いけど。

◀占部　哲彦　広島県
そんなもの持ってお出かけする人っているのでしょうか？　いたら教えてね。でも，せっかく返してもらっても，一度水の中に落ちては……。

◆とある電気店で「遙かなるオーガスタ」を見かけて，「おっ，懐かしいなぁ」と思って手に取ると，なんと500円。思わず買ってしまいましたが，嬉しいような，悲しいような。

津村　忠蔵(19)佐賀県

先のことを考えるとちょっと悲しい……。

◆前から電話は長いほうだったけれど，このところその傾向に拍車がかかってきて，大した用事もないのに，1時間電話しているということが珍しくなくなってしまいました。通信もやっているけど電話にはまた別の魅力がありますね。反応が即座に返ってくるし，それに相手の声が聞けるというのは非常によい。ボイスメールという手もありますが，日本ではまだまだですしね。それにしても，話題が変化して脱線したり，いつのまにやら別の話題になったりする過程には興味深いものがあります。テープにとって分析したら面白いでしょうね。一対一の対話でもそうなのですから，多対多の討論などは優れた司会がいないとめちゃめちゃになってしまうのがわかる気がします。それはともかく，この間，広島－京都間で3時間電話をしたあとは，月末の電話代の請求書を見て顔が青くなりました。DDIか0088に入らないと……。

児玉　茂昭(23)京都府

余計なお世話ですが，友達は近くに作ることをお勧めします。毎月の請求書でスリルを味わっているのでしたらかまいませんが。

◆この夏，体重が約3kg減って45kgになってしまいました（ちなみに身長は170cm）。ただでさえ少ない体重なのに，そのうえ太らない体質のため，もとの体重に戻すのは大変です。私は，いままでに50kgを超えたことがないんです。中学生のころからほとんど変わりがないので……。こういう体質も困りものです。

井上　政宏(24)埼玉県

冬がとっても寒そうですね。

◆Oh! Xの定価が1,000円を超える日は近い。

吉田　征二(21)宮城県

それは超えてもOKっていうことですか？

◆いまでも秋葉原駅で迷う人っているんでしょうか。

及川　恒平(19)東京都

某ライターの友人が初めて東京に遊びにくるとき，待ち合わせの場所が秋葉原駅。うーん，さすが一般人とはちょっと違いますね。

◆とうとう20歳になってしまったー。だけど，特に新たなことはない。編集部の方で20歳のときにすごいことがありましたか？

阿部　嘉(20)宮城県

そりゃもう「ふぇふぇうぇうぇ」で言葉に出してはとてもいえません。

◆仕事帰りの1コイン。生きててよかったと思う。あまり，学生の頃と変わってません。

岩瀬　貴代美(22)福岡県

束の間の幸せを感じるときは人それぞれですよね。皆さんはどんなときに感じます？

◆実家から近い渓谷へ泳ぎに行きました。水の中へ体を沈めてジッとしていると，小魚が体中をつついてくるので笑いを堪えるのに必死です(笑)。それを見ていた周りの友人たちは「あんた……なにしとんのよ」。うーん，なにをしているんでしょうかねぇ，私は。

堂領　輝昌(20)神奈川県

幸せにはいろんな形が……。とはいえ，あまり怪しいことにハマリませんように。

▼吉田　淳一　宮城県
再び登場のPRO子ちゃんですが，彼女の幼児体型にはそんな理由があったんですね。いつの日にかナイスボディになれることをお祈りしています。

▼高田　裕則　愛知県
一生懸命走っている彼女の目指しているゴールはどこなんでしょうか？　それに両隣にいる虫たち（？）は応援，それともライバル？

◆“How are you doing？”がアメリカではよく使われています。中学校で習ったのは”How are you？”なのに。

中村　亮(26)奈良県

挨拶なんてものは時代とともに変わるものです。日本だって「どうも，どうも」ってこれで挨拶になるんですから，これに比べたら……。

◆「リッジレーサー」とか「デイトナUSA」とか，どんどんリアルになっていくと，車の免許を取るときに危なそうである。友人は教習所のオートマ車に乗ったとき，左足でブレーキを踏んで怒られたそうだ。

菰田　英和(24)奈良県

熟練すれば問題ありません(？)。事故を起こしても責任は負えませんけど。

◆ある日，住宅街を流れる用水路の横を通ると猫の声がしました。見ると子猫2匹が水路の底で助けを求めているではありませんか。幸い水深は10cmもなく子猫でもなんとか足がついていますが，水路自体は1m以上の深さで自力脱出はまず不可能です。私は金網を乗り越え水路の中へ飛び降り，生後1カ月ほどと思われる2匹を助けました。その後，公園の木陰に住みついた子猫たちに，悲劇は繰り返されると知りつつ，毎日エサをやってます。

中島　民哉(24)埼玉県

2匹の子猫が無事に育つといいですね。

◆「危険度100％スーパーパワーアップX68000改造大作戦」みたいなものをやってくれると私は嬉しい。

吉田　淳一(21)宮城県

「危険度100％」ということは，絶対にうまくいかないってことのような気が……。

◆99を2乗すると9801になることに気づいた。意外と洒落たネーミングだったことにビックリ。さらに99.1を2乗すると9820.81。四捨五入すると9821に，なんだか悔しいぞ。負けるなシャープ！

大林　光明(18)東京都

「そんなの偶然じゃん」と思いつつも，ちょっとうらやましいかも。

◆9月号の表紙を見ながら次の文章をお読みください。

(女)「う～ん，う～ん，う～ん，う～ん」
(男)「な，なんだ……。うお～，お化け～」
(女)「よくも我々をいたぶってくれたな～」
(？)「ベキ，ボコ，ボカン」某二次元ゴルフ練習場より！
「売り文句：必ず上達してお帰りいただけます」

鈴木　宏良(19)埼玉県

そうか，あのへこみはディンプルだったんですか。いろいろ考えるものですね。

◆〈かれこれ7年前の友人との会話〉

S「私の彼，パソコン買うんだって」
Y「なにを買うの」
S「X68000だって」
Y「でも大学はPC-9801を使ってるんでしょ」
S「そうなの」
Y「じゃあー，機能がそう変わらないのなら，PC-9801のほうがいいんじゃない」
S「あっ，やっぱり」
Y「うん，だって68,000円はちょっと高いよ」
S「私も彼も下宿生だし，やっぱ節約しなきゃね。よーし，9,800円のほうを勧めてみよっと」

ちなみに彼はちゃーんとX68000を買いましたョ。

森本　幸子(28)広島県

懐かしい若き日の思い出ってやつですね。あ，もちろんいまでも十分若いですけど。

◆「手品師」だということがバレて，近所のコンビニのネーちゃんに，小銭は2回，札は3回，チェックされるようになった。

北川　亮(24)東京都

返してもらったお札を確かめるふりをして，1枚「パーム」してしまいましょう（冗談）。わからない人はごめんなさい。

◆今年の夏はすごい暑さで生まれて初めてしたたるほどの汗をかいた。黒い服を着て汗をかいた日の夜，脱いでみると白い模様ができている。……これは汗の塩分か！　ん一東京の夏はすごい（私は青森出身）。

木島　智(24)東京都

かいた汗をちゃんと溜められれば，そこから塩を取り出すことは十分できそうです。まあ，そんなことする人はいませんよね。

◆以前から我が家にはいろいろな生き物が住みついていましたが，今年から新たに狸が加わりました。確かに田舎ではありますが，テレビで見たときは他人事だと思っていたのに妙な気分です。しかし，どこから来たのか知らないけど，予想もしなかった来訪者に，あまりよくないことが続く我が家も少しは救われた気分です。

藤原　彰人(24)岡山県

その狸が幸せを運んでくるといいですね。

◆路上教習中に，一般路で80km/hのスピードで走るのは，なかなか面白い（ただし，先生が半分以上寝ているとき）。

大島　大介(18)北海道

誰か教習所内で80km/hのスピードで走ってくれませんか（そんな奴いないって）。

◆お兄ちゃんにOh!Xを買いに行かされるようになって3年が経過しました。なのに私，まだパソコンのこと，よくわかりません。私，ゲームオンリーだから……。ところで，女子高生の読者ってどのくらいいるんでしょうか……。P.S. チン=シンザンが醜いなんて，ヒドイ。あんな，おちゃめなキャラほかにいないわよ！

綿引　一代(15)茨城県

女性の読者が2％以下，なおかつ女子高生となると1％は切っていると思います。

◆自転車盗難の話，まだまだ甘いですね。私の友人は，東京で盗まれて連絡があったときには新潟からだったそうです。当然，取りには行かなかったようですが，こういうのがゴミになるんですよね。そういえば，明日受けるZ市の職員採用試験の小論文，昨年はゴミについてだったことを思い出してしまいました。こんなことをしてていいんだろうか，2度目なのに……。

中島　太郎(22)神奈川県

太平洋側から日本海側への移動ですが，盗難自転車は果たして海を越えたことはあるのでしょうか？　特に津軽海峡，ちょっと無理かな。

◆少しは役に立つかもしれない，自転車を盗まれにくくするコツ。
1．カギはチェーンロックにする。
2．かごはつけない。
3．駐輪場ではほかの自転車の間に詰めて止める。
理由は各自研究してください。

久米　豊信(26)大阪府

◆自転車盗難のことが載っていたので，地方から大阪に出てくる方は参考にしてください。うちの大学では自転車盗難どころか，よく自販機が壊されて金が盗まれます。これは24時間いつでも誰でも大学に入れるのが原因だと思いますが，大阪は自転車盗難率が全国一です。鍵をかけ忘れたら，駅前なら100％盗まれますし，鍵1個では盗まれる確率は(特にいい自転車の場合)かなり高いです。鍵2個にすると手間がかかるため狙われにくくなります。でも，普通の鍵ならすぐあけてしまう奴はゴロゴロいるのでカード式とか番号式のほうが安心できるでしょう。しかし，自転車関係の業者で軽トラックで盗みに来る極悪な輩もいるので，ワイヤー式のものでフェンスなどに留めてしまうとまず盗まれないでしょう。

中川　直之(22)大阪府

平和といわれる日本でもなかなか大変なものです。気になる方は参考にするといいかもしれません。

◆さぶいっす。どうもうちの上司と私とでは体感温度が違うらしく，クーラーの温度設定の攻防が続いています（ここの営業所は2人しかいない）。お願いだから，22度で強風に設定するのはやめてください。

藤原　常雅(23)神奈川県

悲痛な叫びは届きましたか。でも，これが載る頃は暖房の季節でしょうか？

◆近所の西友の商品搬入口に停まっていた大型トラックのダッシュボードにOh!Xが乗っていた。なんだか，なんだかちょっと，嬉しかった。でも，あれは運転しながら読むんだろうか？

鈴木　幹雄(34)神奈川県

きっと，休憩のお供かと。ありがたいことです。感謝感謝。

◆コンビニでお弁当を温めるとき，「1個で1分のものを2個同時に温めるときは2分」ってオーナーはいうんだけど，私は違うような。だって，あきらかに1分のときよりも熱くなるし，醤油の入れ物は破裂するし……。

林　直貴(22)新潟県

くれぐれも火傷をしないように気をつけてください。

◆「ぷよぷよラムネ」についてくるおまけ（？）のぷよぷよを液体窒素につけて冷やすと，それは見事に爆発します。皆さん試してみてください。

福知　健(23)京都府

よい子の皆さんはまねをしないようにしてください。でも，液体窒素って簡単に手に入りましたっけ。

◆回転寿司があるのなら，拡大寿司や縮小寿司というものがあってもいいはずだが，わたしゃ見たことないよ。

大久保　明弘(22)岩手県

でも，縮小寿司ではきっと食べたような気がしないでしょうね。

◆海に行きすぎて，短期間に4回ほど脱皮したんですが，海パンの下の日焼けしていない部分と段差ができないかどうか心配です（皮4枚分）。

長崎　洋(25)栃木県

日焼けの境目がわからなくなるとき，冬がすぐそこまで来ていることでしょう。

◆盆が過ぎてからは，少しは暑さもやわらぎましたが，それでもまだまだ暑いですね。おまけに，雨が降らない。まだ長崎市内では給水制限は出ていません。洗車をしたいのですが……。昨年は，米不足，今年は水不足。はてさて来年はなに……。

橋本　悟(25)長崎県

うーん，カネ不足。それは，毎年のことでした。失礼しました。

◆大学を卒業後，ゲームメーカーに就職しました。これで，やっとこさ憧れのプログラマとして社会人のスタートが切れたと思いきや，4月初めの健康診断で引っ掛かり即入院。予想外の病態が次々と起こり，入院が長引いてしまいました。ついに7月末には新入社員という立場の弱さからクビになってしまいました。まだ退院の見通しさえ立っていません。不安と焦りのなか，思考はいくつもの無限ループを構成して増殖しています。早く脱出したいですね……。
P.S. 病院で付録ディスクは悲しいです。

森口　明彦（24）大阪府

早く退院できるように祈っています。退院したら，付録ディスクもしっかり使ってくださいね。

◆夏合宿で脱水状態になり，全身けいれんを起こしてしまった。しかし9月号に書かれていた「よこしまな心」のせいではない。ところで，男が女を求めてどこが悪い。男を求める異常でないだけいいと思う。

松嶋　竜(24)東京都

「よこしまな心」のせいでないとすると，本厄の第1弾といったところでしょうか。

◆毎月イラストを出すように心がけているのですが，気がつくと毎月10日過ぎで間に合わないこともときどき（いつも）ある。一応，来年の「ちゃだワ」でランキングを狙っているのですが……（無謀）。

鈴木　道明(25)埼玉県

鈴木さんにはファンレターも届いています。基本的に季節ネタでなければ，翌月分に入るのでぜひ送ってくださいね。

◆今年の夏，コンピュータってのは意外とタフなんだと実感。ちなみに私自身の動作温度の上

◀岩瀬　貴代美　福岡県
「魔法大作戦」ですね。発売日がちょっと延びたのが残念ですが，その分出来がよくなることを期待しましょう。ところで一番目に好きなのは？

◀加藤　信夫　宮城県
この顔のイラストは加藤さんの自画像ですか？　でも，念を押さないと苦虫を噛み潰しているのかと思ってしまうところでした。

限は30℃くらいです。　金沢　輝一(20)東京都

そうすると，今年の夏はほとんど家のなかでクーラーとともに生活していたのでしょうか。活動できたのは夜だけ？

◆名古屋の街を歩いていたら，「美容室さりん」というのを見つけてしまいました。どうしたらいいんでしょうか？　同日，新聞記事のなかに「ナメコオーナー募集」なるものを見つけた。あぁ，なんて日だ。　大塚　京吾(25)岐阜県

美容室に入ってレポートをお願いします。横棒が1本落ちてしまっただけで，本当は「きりん」かも。

◆合宿で運転免許を取りに行ってきました。そこにスーパーファミコンを持ってきた奴がいて，ソフトに「スーパーストリートファイターII」が！　8人トーナメントが面白い。1人くらいコンピュータを入れるとよりグッド。それに勝ち抜き戦も面白い。特にキャラクターを自動で選ばせると，ギャグのような組み合わせがあったりして，それだけでも笑える。おかげで勉強が全然できず，ドイツ語の試験に向けて……Oh!Xを読んでいる。　笹田　泰治(20)京都府

X68000版にもトーナメントモードがありますが，これを利用して来月号ではOh!X編集部内トーナメント大会の模様を紹介する予定です。

◆石油ストーブを組み立てるアルバイトをしています。これを書いているのは8月です（笑）。日中の温度が34℃前後の時期に十数台のストーブがついている様子は異様です。もっとも，クーラーがついているのでそれほど暑くはありませんが。初めてのアルバイトにしては少しキツイとも思いますが見返りもそれなりにあるので……。バイト代でMJ-700V2Cを買おうとしているので，瀧氏の記事に期待しています。

藤谷　武志(19)群馬県

商品として売られるのは寒くなる少し前だから，作られているのは当然なのですが，いまはともかくクーラーがなかった頃はどうしていたんでしょう。

◆PM9：00，ヘトヘトで家に帰り，飯を食いほっとひと息。X68000の電源に指を近づけると嫁さんの目がキラリ！「早くパソコンの楽しさ覚えて通信に手を出そお！」が私のいつものひと言。「ダ〜メ！」が妻のいつものひと言。なんとかしてくれ〜。　久保　英宣(23)三重県

主導権が握れる日がくるといいですね。

◆今年の夏は北海道へ行ってきました。8月の頭から下旬までです。避暑の予定が全然涼しくなく，ようやくこれからという頃に帰途についたわけですからなにか割に合わない気がしました。それでも35℃は超えませんでしたからよしとしますか。　西川　和範（20）東京都

同じ30℃でも北海道と東京では湿気が違うので……うらやましい。

◀平　智征　神奈川県
見つめ合う2人の天使(?)。この天使たちが交わした約束がとっても気になるところですが，もしかしたら，妖しい関係？　ごめんなさい。

◆「ジオグラフシール」ですが，1回もダメージを受けずにクリアすると70万点です。1,2,3面あたりでなんとかできます。私はそれで195万点を取りました。ちなみに最速タイムは14分15秒です。13分台はいけそうですが……。あと，「餓狼SP」でリョウ・サカザキをストーリーモードで使える方法があったら教えてください。誰か探して。　金渕　満(17)青森県

「ジオグラフシール」のシークレットボーナスはこんなところでしょうか。そのほかのゲームの情報もお待ちしております。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

売ります

★東京システムリサーチのメモリボード「Xsimm10」と4MバイトSIMM（70ns）を2枚セットで38,000円くらいで売ります。箱，説明書はあります。連絡は往復ハガキでお願いします。〒146 東京都大田区池上1-23-10　金井利成(22)

★カラーイメージジェット「IO-735X」を50,000円(送料込み)で売ります。箱，付属品，説明書などすべてあります。手渡しできる方を優先します。連絡は往復ハガキでお願いします。〒970 福島県いわき市平下平窪字白山下12　高田英夫(28)

★X68000 Compact XVI用2MバイトRAMボード「CZ-6B E2D」（コプロMC68882付き）を25,000円，東京システムリサーチのメモリボード「Xsimm10」(4MバイトSIMM2枚付き)を40,000円で売ります。どちらも箱，付属品などあります。連絡は往復ハガキでお願いします。〒350-13 埼玉県狭山市狭山台2-24-104　木下卓也(22)

★Macintosh用72ピン8MバイトSIMM（60ns）1枚を送料込みで27,000円で売ります。連絡は往復ハガキでお願いいたします。〒131 東京都墨田区八広1-8-4　折田貴弘(22)

★ローランドの音源モジュール「CM-64」を送料込みで45,000円で売ります。ほとんど使用していません。箱，付属品などすべてあります。連絡は官製ハガキか封書にてお願いします。〒260 千葉県中央区葛城1-3-9　駒井健一(23)

★X68000XVI用2Mバイト増設RAMボード「CZ-6B E2A」を15,000〜20,000円で売ります。箱，説明書もあり新品です。値段の高い方を優先します。連絡は往復ハガキでお願いします。〒961 福島県白河市女石15-7　斉藤　洋(21)

★アイ・オー・データ機器の2Mバイト増設RAMボード「PIO-6BE2-2M」を14,000円（送料込み）で売ります。付属品はすべてありで完動品です。連絡は往復ハガキでお願いします。〒431-33 静岡県天竜市船明280　坪井秀次(23)

★48ドット熱転写カラー漢字プリンタ「CZ-8PC5」を新品リボン2個つきで10,000円で売ります。送料込みです。連絡は往復ハガキでお願いします。〒143 東京都大田区中央6-15-10　今井佑(17)

買います

★東京システムリサーチのメモリボード「Xsimm10」を10,000円で買います。送料込みでお願いします。連絡は往復ハガキでお願いします。〒438 静岡県磐田市今之浦4-7-9サントピア105　高野正文(32)

★カラーイメージユニット「CZ-6VT1」を45,000円で買います。付属品と説明書があれば箱はなくてもかまいません。連絡は官製ハガキでお願いします。〒350-02 埼玉県坂戸市泉町2-3-13コーポ泉202号室　伊藤敏朗(23)

★SCSIボードを13,000円で買います。ケーブルもあれば買います。連絡は往復ハガキでお願いします。〒616 京都府京都市右京区太秦安井車道町19-33ピア花園A-5　藤井栄一(19)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は9月号の内容に関するレポートです。

●9月号の特集「SX-WINDOW環境セットアップ」は，正直いってSX-WINDOWユーザーではないのでどう評価してよいものかわかりません。ただ，以前からシャーペン.Xには少し興味をもっていたので「そろそろSX-WINDOWを使ってみようか」と思っています。

しかし，SX-WINDOW関係の記事を読むたびに思うことは「ハードの制約に束縛されている」ということです。もっとも7年前のハードに，トレンドなシステムを走らせようとしているのですから，業界から見れば無謀もいいところなのですが。確かに，その無謀さがカッコイイのならいいのですが，現在のSX-WINDOWを見るかぎり少し不細工な面があるような気がしてなりません。

これは，勝手な思い込みなのですが，「ハードの制約を感じさせない無謀さ」がX68000らしさだと私は思っています。ですから，ハードの制約を感じさせてしまうSX-WINDOWは，少し好きになれないのです。現在SX-WINDOWは「シャーペン.Xだけ」といった雰囲気がありませんか？ もしもそうだったら，私は「もっとカッチョイイ無謀を！」と叫びたいです。

あと，9月号から定価が上がりましたが，もしも私が普通の読者であっても購読は続けると思います。Oh!Xは私にとって数少ない情報源のひとつですから。ただ「パソコン通信をやっていたらどう考えたか？」とも思いますけど。それにしてもシャープはXシリーズをどうするつもりなのでしょう。以前からのシャープマシンユーザーが「歴史は繰り返す」と悲しそうに訴えていますが……。

中矢　史朗(23)　X68030/X68000　ACE-HD, PC-386P　愛媛県

●正直にいうと，定価680円は高いです。現在のXシリーズ環境が厳しいし，そのなかで専門誌を作っているのだから値上げもしかたないかな……と多くの読者は思うでしょう。単純に考えて1割以上値上げしたのだから，現状の9割以上の人が引き続き購読してくれれば，とりあえず値上げの効果があったことになるでしょう。しかし，それは固定読者を少しずつ減らすことになるようにも思います。たかが80円，されど80円です。

少し話がそれますけど，現在，宙に浮いているX68000ユーザーって結構多いのではないでしょうか。つまり，「ゲームは出ないし，ほかのパソコン誌じゃ名前も見かけないし……」そんな人たちを捕まえるチャンスだと思います。いままで，ゲームしかしてこなかった人たちって，たぶんメインメモリ2Mバイト，ハードディスクなしの人が多いことでしょう。そんな人たちのクリエイティブマインドを刺激する本作りをしていけば，逆に読者も増えるかもしれませんよ。

そして，9月号で印象に残った記事といえば「怪しいZ80の使い方」です。いいなあ，このノリが。こういうのが俗にいう“Oh!Xらしい記事”なんでしょうね。どちらかというとアプリケーションの利用の記事が多くなってきたOh!X上で，かなり新鮮に感じます。同じようなパターンで68000編もやってほしいものです。

渡辺　祐介(20)　X68000, MSX2　富山県

●SX-WINDOWもver.3.1となり，ver.1.0の頃を考えればずいぶんとまとまってきているので，私としては結構評価しています。これなら，ジャストウィンドウといい勝負，とか思っているのですが，問題は「アプリケーションが少ない」よりも，ユーザーからの「なにか作りたいからフォーマットを公開して」的な声がいまだに多いことではないでしょうか。つまり，SX-WINDOWの抱える問題点は，SX-WINDOW自体にあるわけではなく，いまいちなシャープの姿勢にあると思うのです。

中村　俊之(20)　X68000 EXPERT, X1 turbo model30, MZ-1500/700, PC-E500, PC-286noteF　東京都

●9月号の特集は「SX-WINDOW環境セットアップ」というタイトルがついていますが，内容的には大部分がシャーペンのカスタマイズで占められているように思います。ソフトのカスタマイズは，使用する人によってその方法はまちまちでしょう。結局，カスタマイズの方法を紹介する程度の内容が精一杯ではないでしょうか。実際に今回の特集記事の内容はそうであったと思います。

今回の特集は「SX-WINDOW環境セットアップ」と銘打つだけの内容ではないのでは？と考えさせられます。ウィンドウ環境の特徴は，ソフトによる操作方法の統一化，また，データの共有化ではないでしょうか。今回の特集は，その意図が見えてきませんでした。

壁谷　喜嗣(35)　X68000 EXPERT, PC-9821As/9801NS/E　愛知県

●新製品紹介の「MJ-700V2C」。カラーインクジェットプリンタも安くなったものです。ひと昔前までは，カラー印刷といえば熱転写だと思っていたので。それにしても720dpiモードはとてもきれいですね。しかし，SX-WINDOWから利用できないのが痛いところです(瀧さんに期待か？)。

そして「X68030 D'ash」。なんとなく出るのではないかと思っていました。私から見ればX68030でも相当速いと思うのですが，慣れてしまうと欲が出るんでしょうかね。記事に関しては，結構やさしく書いてあるので安心して読めました。4ページも使ってあるし，テストなどの結果も詳しく書いてあってよかったです。

あと「GENIE」以来CGAに興味をもっているので，DōGAのページはいつも楽しみにしています。そこでオッと思ったのはQ&A特集です。こういった記事は個人的に大好きです。私もなにかに意見することが好きなので，いろいろな意見があるのだな，と思いながら読み進めていきました。やはり意見は大切です。とりあえずOh!Xの読者の皆さんも，つまんないことでもいいからアンケートハガキを出すべきではないでしょうか。せっかく，切手を貼らなくてもいいようになっているのですから。

大上　幸宏(21)　X68000 PROII　鹿児島県

ごめんなさいのコーナー

10月号　特別企画「もみじ狩りPRO-68K」

付録ディスク「ベル.X」で，サブウィンドウを「取消」で閉じると，二度とウィンドウが開かなくなるバグが発見されました。

あとウィンドウデザイナでも不都合が発見されています(詳しい症状は102ページを参照してください)。それぞれ訂正用の差分リストが，54ページに掲載されていますので，付録ディスクに収録されていたリストを修正してください。

バグに関するお問い合わせは
☎03(5642)8182(直通)
月〜金曜日16：00〜18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

なにはなくとも基本はBASICかな?

▶誰でも気軽にプログラミングの世界へ招待してくれるX-BASIC。やはり標準添付の強み，とでもいうのでしょうか，実際にかなりの人がX-BASICを使ったことがあるようです。

それに，手軽にX68000の世界に触れたければ，X-BASICを使うのが一番お勧めです。なぜお勧めかというと，レスポンスの問題。つまり，X-BASICはインタプリタですから，実行しようとする答えがリアルタイムに確認できるからです。やはり，やってみようとすることが，即座に確認できるというのは嬉しいことですからね。

もちろん，インタプリタということで，実行速度がおぼつかない場合があります。そんなときには，BAStoCでコンバート。あっという間に「.X」の実行形式のファイルが出来上がり！　そんなコンパイル環境も整っているのもユーザーにとっては嬉しいかぎりです。

そして，X-BASICのもうひとつの魅力は，外部関数をユーザーが自由に作成できることです。これによって，X-BASICがさらに進化できるようになっています。いままで，Oh!Xでもさまざまな外部関数が発表されましたし，汎用性はないものの専用の外部関数を整備することで，いろいろなゲームも作られてきました。つまり，不満があれば自分でなんとかできる仕様となっているのです。

プログラミングを始めたいな，と思ったら，X-BASICを立ち上げてみてください。そして，X68000の魅力あふれる世界を体験してみましょう。

▶さて，予告に何度も登場しながら，読者の皆さんを裏切り続けていた「H.A.R.P」がようやく編集部に届きました。12月号では絶対にレビューを行いますので，楽しみにしていてください。

▶「ファイル共有の実験と実践」は著者多忙のため，「X68000マシン語プログラミング」は，著者多忙+スランプ脱出ならず，につきお休みとなってしまいました。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたフロッピーディスクを添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒103 東京都中央区日本橋浜町3-42-3
ソフトバンク出版部
Oh!X「テーマ名」係

SHIFT・BREAK

▶大学の研究室に入れるIBMの互換機を買いにいった。CPUは486DX2-66で，お店で組み立てて送ってもらうようにしたのだが，なかなか届かない。3週間ほどしてやっと送ってきたのだが，動かしてみるとなんか速い。開けてみるとDX4-100が載っている。遅れたお詫びだと思っておこう。インジケータは66のままだったけど。（I.K）

▶知り合いのパソコンショップの店員の話では，どうもX68030が手に入りにくくなってきているらしい。NewXが出る前に生産完了なんて絶対やだぞ。でも，もしそうなら……。とにかくX68030の価値がわかっている人なら25万円前後の値段は高く感じないはず。いつかはX68030と思っている人たちは，借金してでもいま，X68030を！（瀧）

▶周りのお膳立てとはいえナイナイが独走体勢のままの「とぶくすり」再開。ほかのメンツとのぎくしゃくした関係がいまから楽しみだ。成子坂がブレイクしそうでしないのは見ていても歯がゆいが，ライブネタの健在ぶりがまだ闘う意志を感じさせてくれる。ローカルライブでもいい新人が結構出てきているし，お笑い界からはまだまだ目が離せない。（哲）

▶テレビを買おうと思い秋葉原へと出かける。どうせ買うならばいろいろと応用が利き，ゲーム機の映像もきれいに映るRGBマルチ端子つきがいいと探す。しかし文字放送ブームが終わったいま，この端子つきのモデルはソニーから数機種出ているにすぎないという。あって損はない端子だと思うんだけどな，家電メーカーさん。（善）

▶先月のpTeXの件の続きです。X68000/030のpTeXで縦書きは「できます」。ただTeX本に収録されている縦書き文書のサンプルがある特殊な条件を満たす必要があるのでプレビューアがこけたのです。いずれにせよご迷惑をおかけしました。この件がもとで，より安定したドライバを作っていただけることになったのが救いといえるでしょう。（A.T.）

▶最近雑誌ばかり読んでいる。パートは止めたが，代わりにマルコ・ポーロやダ・ヴィンチが加わるし，SPA!と噂の真相と週刊サッカー・マガジンは欠かせない。特選街は送ってくるし，ジャンプ，サンデー，マガジン，スピリッツ，オリジナルは惰性で続けているし，これに10数誌のパソコン雑誌が加わる。しかし，なにか物足りないのだ。なんだ？（K）

▶出勤できるようになったと思ったら自宅療養の生活に逆戻り。どうもひと筋縄ではいきそうにない。暇なので「マザー2」をやる。「はえみつ」を使わないと敵に勝てなかったり，「いのる」ことをしないとゲームを終われないのは不条理だ。ヒントはそれとなく出ているのだがまわりくどすぎる。ジュースをこぼして取り合うほどのゲームだろうか。（KO）

▶アジア大会の日本のサッカーはいったいなんだ。どうにもひどい。カタール戦を見たあとの深夜，フジテレビで「セリエAダイジェスト」をやっていた。あんなプレイを見せられたらJリーグの試合なんて見てられない。でも，みんながそう思えば，客の入りも悪くなって生で見やすくなるかも……。ただ，プレイの質が……やっぱりイタリアか。（高）

▶人間がとろいあたしでも，ぬるい難易度ならとりあえずエンディングが見られたスパIIだが，春麗が出てくると，どうしてもハマりはまぬがれない。防御ができず（！）気功拳で削られるのもさることながら，いちばんの問題は女の子相手だとどうも闘志が湧かないことにあるようだ。ちなみにキャミイで闘って，最後だけキャラを代えるんだけど……。（ふ）

▶10月から秋のテレビ番組改変ということで，僕の好きだった「ポンキッキーズ」も朝の「ポンキッキ」に吸収されてしまい，ちょっと残念。あのわけわからんポンキッキーズ体操とか，花子さんシリーズはよかったのに。でも，深夜番組の「テレコンワールド」が生き残っているからいいや。（スキューバまでこなすガチャピンはすごい，と尊敬するJ）

▶某誌に載ったシャープ辻社長の話によると1年後に「ひと味違うパソコン」が出るという。ハードウェアスペックではそれなりのマシンを出すとは思うのだが，記事を読む限り現状把握度にちと不安がある。いまさらマーケティングに期待してるわけではないが……。しかし，さらに1年てのはちと長いかもしれない。（U）

▶パソコンのコマーシャルネタ第2弾。「Think Pad, Think Pad～♪」って歌うIBMのテレビCMなんだけど，英語で正しく発音されると日本人には「ティンクパッド」って聞こえちゃう。と思っていたら，なんと「シンクパッド」ってものすごくわざとらしい発音で歌うバージョンが出てきた。宣伝部のお偉いさんが文句でもつけたのかなあ。（T）

microOdyssey

非常に個人的なことなのだが，最近ちょっと元気回復に役立ったことについて書こう。

故あって部屋の大々的な片付けをしていたら，デッサンやらエスキースやら言葉の断片などを書き散らしたものやらが，いくつか出てきた。学生時代からちょっと前までのもので，すべてが残っているわけではないが，突然出てきたそれらを見ていて，いまの私はちょっと恥ずかしくなった。もちろん，その内容についての心持ちもあるが，それより何より突きつけられたのは，最近の私のナマケモノぶりだ。それらを見ると，最近いかに毎日を意識的に過ごしていないかがわかる。忙しさを自分への言い訳にして何も形に残してはいないし，見聞きした情報を自分のなかにきちんと蓄積しているのかどうかもはなはだあやしい。手を動かしていないので，出合ったものごとをきちんと咀嚼せず，消化不良を起こしていることに気がついた。

青臭い文章や下手ななぐり描きをいくつ連ねても，そのもの自体には価値はない。他人の心を動かすほどのものではないし，完成した作品ですらない。ただ，「制作」というのとは別の次元で，その紙と筆記具による「行為」は私にとっては意味がある。

メモをとることで，その対象物や出来事をきちんと把握する。絵や図(たとえ落書き程度のものであっても)を描くときには，ふつうなら気にも留めないような細部をよく見つめ，その対象物の内部や構造までに考えをめぐらせる。そういうふうにして，私はいろいろなものを見たり聴いたり感じたりしてきたのだ。

それは，自分以外のものに限らず，自分自身の感情や感覚についても同様だ。たとえ断片でも自分の外に出すことで発見をし，理解して，認識が生まれてくる。だから私は描くために見るのではなく，見るために描くのだ。

もちろん，とりたてて何かを書いたりしなくても考えることはできる。また，描くことをしなくても対象物をよく見ることはできる。ただ，書くことや描くことは意識をもつための手助けになる。そして，どうやらいまの私は，書くように考えるとか描くように見つめることを怠っていたのだということを，昔のスケッチブックなどが教えてくれたのだ。記録が目的ではなかったので，いま見ると意味不明のものもたくさんあるが，そこには，何かをいつもきちんと見つめていた自分があった。過去を懐かしむというのではないが，昔の自分に「しっかりせい」と励まされたようで，これからは，ときどきは昔の自分と向き合って，忘れてはいけないものや捨てるべきことなどを考えようと思ったのだ。

で，実をいえば，私にとって表現が認識のための方法だとはいえ，ささやかな夢もないわけではない。創作への努力を半ば放棄しておきながら非常に甘ったれた考えだが，生きているうちに1枚でも，気に入った絵が描けたらいいな，なんて思ってもいる。別に絵画でなくてもいい。形は何であれ自分自身を表出できれば，他人には理解してもらえなくてもいい。これこそが自分の内部に在る，というものを出すことができたら……。もちろん，それでもし共感してくれる人がいたら，それは幸福の極みだろう。

そんな思いも，また鮮やかさを取り戻してきた。いつか，を夢見てがんばろうかな。　(ふ)

1994年12月号11月18日(金)発売

Oh!X創刊7周年記念特別企画
Oh!X懐ゲー制作工房
特別付録　5″2HD「XL/Image」体験版＋α
THE SOFTOUCH SPECIAL
Oh!Xライター対抗スパII対戦
新製品紹介　H.A.R.P(今度こそやります！)

予価　900円

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートBI	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3209)0656
	渋谷	大盛堂書店	03(3463)0511
	池袋	旭屋書店池袋店	03(3986)0311
	八王子	くまざわ書店八王子本店	0426(25)1201
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	043(224)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある「新規」「継続」のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

基本的に，定期購読に関することは販売局で一括して行っています。住所変更など問題が生じた場合は，Oh!X編集部ではなくソフトバンク販売局へお問い合わせください。

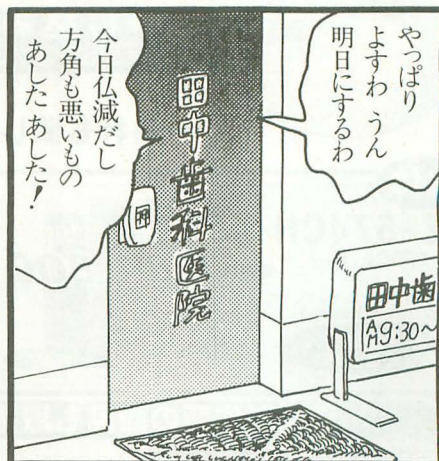
海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

11月号
■1994年11月1日発行　定価680円(本体660円)
■発行人　橋本五郎
■編集人　稲葉俊夫
■発売元　ソフトバンク株式会社
■出版事業部　〒103 東京都中央区日本橋浜町3-42-3
Oh!X編集部　☎03(5642)8122
販売局　☎03(5642)8100 FAX 03(5641)3424
広告局　☎03(5642)8111
■印　刷　凸版印刷株式会社
©1994 SOFTBANK CORP. 雑誌02179-11本誌からの無断転載を禁じます。
落丁・乱丁の場合はお取り替えいたします。

満開の電子(でんこ)ちゃん

作・え 岡村祭(おかむらまつり)

77号(9/18発売)には他にも、MSXLIBプログラミング講座、レイトレ講座などの連載陣、GRW対応Pi表示ツールなど、2枚組でボリュームアップ!

講読方法:定期購読もしくはソフトベンダーTAKERUでお買い求めいただけます。
★定期購読の場合=購読料第76号(94年9月号)より6ヶ月分9,000円(送料サービス、消費税込)を、現金書留または郵便振替で下記の宛先へお送り下さい。
　現金書留の場合:〒171 東京都豊島区長崎1-28-23 Muse西池袋2F ㈱満開製作所
　郵便振替の場合:東京 5-362847 ㈱満開製作所
●ご注文の際は、郵便番号・住所・氏名・電話番号を忘れずに記入して下さい。
●3.5インチディスク版をご希望の方は、「3.5インチ版」とご指定下さい。
●新規購読の方は「新規」と明記して下さい。なお、特に購読開始号のご指定がない場合は既刊の最新号からお送りいたします。
●製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しします。
★TAKERUでお求めの場合=1部につき1,600円(消費税込)です。
●定期購読版と内容が一部異なる場合があります。御了承下さい。
●お問い合わせ先 TEL(03)3554-9282(月〜金 午前11時〜午後6時)
(なお、定期購読版のバックナンバーについては定期購読の方のみご注文を承ります)

北野 雅利(大阪府)

毎度。最近は、パソコンちゅうたら窓が付き物らしいでんな。扉とかの類とはちゃいまっせ。かっこよお言わせてもおたら、ウィンドウですわ。そのウィンドウ扱お思たら、鼠が必要らしいですな。無論生きてる奴とは違うてマウスとかいうけったいなもんですな。「残念やけどX68持ってへん」という方もよお聞いておくれ。電クラは「マウスひとつでラクラク操作」が売りですねん。そんなあんさんでも、マウス捌きの勉強になりまっせ。満開はんではX68も通信販売してますさかい是非一緒にどうでっか?

“冬のボーナス先取り!! 10/21(金)～11/20(日)まで開催中!!”

TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO

お申し込みは今すぐ！ 受注専門フリーダイヤル 0120-377-999

～あれも欲しいし、これも欲しい、み～んなまとめてボーナスで買っちゃおう!!～

X680x0シリーズ

本体

CZ-674CH(X68000 CompactXVI)

超特価 **¥84,800**

TS-XFDCAを使えば、縦置き5インチモデルX68000シリーズ(PROシリーズを除く)を外部ドライブとして使用可能！

是非、2台目のマシンとしてどうぞ！

※モニタは別売です

お勧めのセット

CZ-674CH・・・・ ￥298,000

CZ-608D-BK・・ ￥ 94,800

ツクモ特価￥150,000

—— X68030 ——

お勧めの組み合わせ!!

CZ-500C-B・・・・・ ￥398,000

290MBハードディスクサービス

ツクモ特価￥285,000

※モニタは別売です

満開製作所の商品も取扱中！

X68000 CompactXVI 24MHz改

RED ZONE………………………… ツクモ特価￥ 98,000

RED ZONE(2DD)…………………… ツクモ特価￥103,000

満開製外付け5インチFDD

MK-FD1…………………………… ツクモ特価￥ 39,800

MK-FD1(カラーリングモデル)……… ツクモ特価￥ 44,800

X 680x0シリーズ用RAMボード

	ツクモ特価
SH-6BE1-1ME(CZ-600C専用)……	¥10,500
PIO-6BE1-AE(ACE/PRO/PRO2シリーズ用)……	¥10,500
PIO-6BE2-2ME(拡張スロット用)……	¥22,500
PIO-6BE4-4ME(拡張スロット用)……	¥38,200
SH-5BE4-8M(X68030シリーズ用)……	¥44,000
CZ-6BE2A(XVI専用)……	¥42,500
CZ-6BE2D(CompactXVI専用)……	¥29,800
TS-6BE2B(CZ-6BE2A/D用拡張RAM)……	¥29,800
X SIMM10-8M(拡張スロット用8MB)……	¥53,800
X SIMM10-10M(拡張スロット用10MB)……	¥64,800

X680x0ユーザーの為のツクモオリジナルシリーズ

TS-3XRシリーズ X680x0用外付けドライブ

●2DD/2HD/2HC/1.44MBフォーマット対応
※2DD/2HC/1.44MBを使用するにはHuman68K Ver.3.0以上が必要
●CompactXVI/68030用ケーブル付

TS-3XR1B 1ドライブ 定価¥33,800…………
ツクモ特価¥26,800

TS-3XR2B 2ドライブ 定価¥46,800…………
ツクモ特価¥36,800

簡単コンピュータミュージック

Music Card for X680x0(TS-6GM1)

音源を搭載したMIDIボードの登場。
これ1枚で手軽にMIDIコンピュータミュージックが楽しめます。
GM規格・MT－32・CM－64等の音色配列をサポート。
最大同時発音数16。「Mu－1GSお試し版」付き

ツクモ特価¥24,800

オリジナルRAMボード

	ツクモ特価
TS-XM1-4A(拡張スロット用4MB)・・・・・・・・・	¥39,800
TS-XM1-6A(拡張スロット用6MB)・・・・・・・・・	¥47,800
TS-XM1-8A(拡張スロット用8MB)・・・・・・・・・	¥55,800
TS-XM1-10(拡張スロット用10MB)・・・・・・・・	¥63,800

NEW *X68000Compact/RED ZONE用内蔵6MB+FPUボード*

TS-6BE6DP ※FPUにMC68882を使用しているため、Human Ver3.0より前に付属していたFLOAT3.Xでは使用出来ませんのでご注意下さい。

特価¥57,800 定価￥64,800

オリジナルSCSI&RAMボード

NEW **TS-6BS1mkII** 近日発売予定！

変更点 その1 接続コネクタをフルピッチからハーフピッチコネクタに変更致しました。

変更点 その2 72PINのSIMMメモリソケットを、一つ用意しました。これにより拡張スロット不足でお悩みの方に朗報です。

特価¥39,800

ジョイスティックパラレルインターフェイス

拡張スロットを使用しません。ジョイスティック端子に接続できるパラレルインターフェイスです。
これでスキャナーも高速で取り込みが可能になります。
★取り込みソフトェア及びサンプルソースが付属致します。

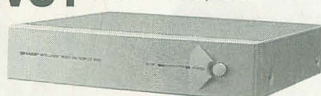
TS-JPIFS(CZ-8NS1対応用) 定価￥17,800………… **ツクモ特価￥14,800**

TS-JPIFE(EPSONスキャナー対応用) 予価￥17,800 NEW **ツクモ特価￥14,800**

映像関連機器

動画を始めてみませんか？

ビデオ入力ユニット **CZ-6VS1** 定価￥178,000

MC68EC020(25MHz)の32BitMPUを搭載し、SCSIインターフェイスを介してパソコンへデータを転送。動画・静止画を簡単に保存出来るアプリケーションソフト「ライブスキャン」を標準装備。1,677万色まで対応し、最大640×480ドットの高解像度で、高速取り込が可能です。但しX680x0シリーズでご使用の場合には6万5千色までの表示となります。

ツクモ特価￥142,000

多機能対応型スキャンコンバーター

電波新聞社 **XVGA-1v** 定価￥69,800

X680x0シリーズやその他のパソコンの水平周波数(24KHz/31KHz)をNTSC標準信号に変換するスキャンコンバータユニットですので、家庭用テレビやビデオ・デッキで映像を表示または録画することができます。また、ビデオプリンターを使えば画面のハードコピーも可能です。

ツクモ特価￥59,300

ビデオプリンター(昇華型)

シャープ **VP-ES1** NEW

高画質ハガキ大プリント、普通紙・布転写用紙もOK。
4分割、16分割、ストロボも可。
※入力信号は、ビデオ信号となりますので、パソコンに接続の場合にはお問い合わせ下さい。

ツクモ特価￥49,300

カラーイメージスキャナー

JX-330X 定価￥178,000

ADF・透過原稿対応型カラーイメージスキャナの登場です。

高解像度(600dpi)、超高速が特長です。
ScannerTools(画像入力ソフト)付属。

ツクモ特価￥138,000

台数限定 **CZ-8NS1**・・・・・・・・ **ツクモ特価￥69,800**

【東 京】●パソコン本店（各種パソコン・周辺機器）●パソコン本店II（パソコン・ワープロ）●DOS／Vパソコン館（DOS／Vパソコン・下取り）●万世店（総合通信機器）●5号店（ビデオ・ムービー・CS）●ソフト8号店（パソコン＆ゲーム用ソフト）●買取センター（ゲーム機・ゲーム機用ソフト買取り）●ニューセンター店●ラジオセンター店（ハンディーレシーバー・テレホンパーツ）
【名古屋】●名古屋1号店（パソコン全般）●名古屋2号店（パソコン全般・総合通信機器・ビデオ）　【札 幌】●札幌店（パソコン全般・総合通信機器）（パソコン全般）

業界で注目の低金利!! 12回払い、7.5%がナント6%に!

クレジット金利がこんなにお安くなりました! ~月々ムリのないお支払い額で欲しかったパソコンがお手元に!!~

支払回数(回)	1	3	6	10	12	15	18	20	24	30	36	42	48	54	60
分割払い手数料率(%)	2.5	3.5	4.5	5.5	6	9	11.0	12	12.5	16.5	17.5	22	23	28.5	29.5

TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKU

受付時間(平日)AM10:45~PM7:30
(日・祝)AM10:15~PM7:00 木曜定休

『FAX24時間お見積もり受付』 **03-3255-4199**
お名前.住所.電話番号.FAX番号をご記入の上ご依頼下さい。

ツクモグローバルJCBカード

JCBならではの国内・海外サービスにツクモオリジナルの特典をプラス。ツクモ各店にある入会申込書にてお申し込み下さい。くわしくはグローバル事務局03(3251)9898又は各店へ。
※ジャックス・VISA・セントラル・マスターも取り扱っております。

※フェアーのお知らせ

恒例「シャープわんさかバザール」を開催致します。
場所:ツクモパソコン本店II
期間:11月19日(水)~11月30日(日)まで
期間中、パソコン本体をお買い上げの方には今回特製オリジナルグッズ「X68030電飾POPスタンド」をプレゼント!

特価品はお店に来ないとわからない!!

先着 50 名様

ディスプレイも特別価格にて提供中!

CZ-608D(14型カラーディスプレイ) ツクモ特価¥69,000
CZ-615D(15型カラーディスプレイテレビ) ツクモ特価¥132,000
CZ-621D(21型カラーディスプレイ) ツクモ特価¥125,000

プリンター

BJC-400J NEW (ケーブルセット)
モノクロ高速
カラー対応
エコノミータイプ
ツクモ特価 ¥62,800

カラーバブルジェットプリンター
BJC-600J (ケーブルセット) カラー高速印字 スタンダードタイプ
ツクモ特価 ¥76,000

マッハジェットカラー
MJ-700V2C (ケーブルセット)
ツクモ特価 ¥77,000

バブルジェットプリンター
BJ-10VLite(ケーブルセット)......... ツクモ特価¥32,000

大容量記憶装置

MO特選セット
SCSIボードが必要な場合にはセット価格に¥22,000加算となります。

- Logitec LMO-400 (230MB) ¥158,000 プラス MOメディア SCSIケーブル ターミネータ ツクモ特価¥112,000
- Logitec LMO-200 (128MB) ¥79,800 プラス MOメディア SCSIケーブル ターミネータ ツクモ特価¥69,000
- Panasonic LF-3200JD (230MB) ¥168,000 プラス MOメディア SCSIケーブル ツクモ特価¥128,000
- COPAL CS-M230PA (230MB) ¥148,000 プラス MOメディア SCSIケーブル ツクモ特価¥110,000

ハードディスク
290MBハードディスク..... ツクモ特価¥29,800~
350MBハードディスク..... ツクモ特価¥44,800~
500MBハードディスク..... ツクモ特価¥49,800~

CD-ROMドライブ
		ツクモ特価
SONY	CDU-7811....(SONYキャディ)	¥36,800
Logitec	SCD-200......(NECドライブ)	¥17,800
Logitec	SCD-400......(NEC製4倍速)	¥41,800
PIONEER	DR-U104X....(4倍速)	¥64,800
PIONEER	DRM-602X...(4連装2倍速)	¥55,800
AIWA	ACD-300WN	¥29,800

CD-ROMドライバーソフト+SCSIケーブル ¥9,000

MIDIコンピュータミュージック特選セット

Roland SC-55mkII セット
SC-55mkII........¥69,000
TS-6GM1.........¥39,800
MIDI変換ケーブル...¥4,000
合計定価¥112,800
ツクモ特価¥79,800

Roland SC-88 セット
SC-88............¥89,800
TS-6GM1.........¥39,800
MIDI変換ケーブル...¥4,000
合計定価¥133,600
ツクモ特価¥99,000

コンピュータアート スーパーグラフィックツール

その1.慣れてしまうとマウスがいらない
DrawingSlate............¥74,800
Matier Ver2.0............¥39,800
合計定価¥114,600
ツクモ特価¥88,800

その2.ハイクオリティなのにこんなに安い
MJ-700V2C..............¥99,800
プリンターケーブル.......¥4,800
Matier Ver2.0............¥39,800
合計定価¥114,400
ツクモ特価¥110,000

電子文具

これぞ、パーソナルシステムの決定版!!

PI-4000 定価¥75,000 ツクモ特価¥59,800
PI-4000FX 定価¥91,000 FAXモデムセットモデル ツクモ特価¥72,800
ink ZAURUS (PI-4000シリーズ)登場!!

※パソコンリンクソフト SX-WINDOW デスクアクセサリ集 ツクモ特価¥11,800
※パソコンリンク用ケーブル CE-150TS ツクモ特価¥10,200

パソコン通信

モデム
		ツクモ特価
AIWA	PV-BF144	¥19,800
OMRON	ME1414B	¥22,500

通信ソフト
		ツクモ特価
SPS	た~みのる2	¥13,000
SHARP	Communication SX-68K	¥15,800

ソフトウェア

	ツクモ特価
OS-9/X68030 V2.4.5	¥20,000
Technical Tool Kit V.2.4.5	¥16,000
Ultra C & Professional Pack V1.1	¥36,000
X Windows V11.5	¥24,000
SX-WINDOW Ver3.1システムキット(NEW)	¥18,200
SX-WINDOWデスクアクセサリ集	¥11,800
C COMPILER Ver2.1 NEWKIT	¥35,800
Easydraw SX-68K	¥15,800
Easypaint SX-68K	¥10,200
SOUND SX-68K	¥12,600
Communication SX-68K	¥15,800
Matier Ver2.0	¥29,800

	ツクモ特価
CD-ROM Driver	¥4,800
SX-PhotoGallery	¥15,800
DoubleBookin'	¥12,800
SX広辞苑(CD-ROM別)	¥17,800
EGWord SX-68K	¥47,800
SX-WINDOW開発キット	¥31,800
開発キット用ツール集	¥10,200
倉庫番リベンジSX-68K	¥5,400
MUSIC SX-68K	¥30,400
XDTP SX-68K	¥28,000
フォント&ロゴデザインツールSX-68K	11月発売予定
Super BUSINESS	NOW WAITING

秋葉原

名古屋

札幌

お支払い方法

あなたのご都合に合わせていろいろ選べます。

クレジット払い
月々¥3,000以上の均等払いも頭金なし。夏・冬ボーナス2回払いもOK!

カード払い
¥5,000以上
通信販売での御利用カード
ツクモグローバルカード・セントラル・ジャックス
※御本人様より電話で通信販売部へお申し込み下さい。

各種リース払い
詳しくは各店にご相談下さい。

現金書留払い
〒101-91 東京都千代田区神田郵便局私書箱135号
ツクモ通販センター Oh!X係

代金引き換え配達
お申し込みは電話1本でOK!
配達日の指定もできます。

銀行振込払い
事前にTELでお届け先をご連絡下さい。
三和銀行 秋葉原支店
(普) 1009939 ツクモデンキ

商品についてのお問い合わせは各店に

秋葉原
(営)平日AM10:45~PM7:30日・祝AM10:15~PM7:00

ツクモパソコン本店II 4F
03-3253-1899
03-3253-4199(代)
(休)木曜日

ツクモニューセンター店
03-3251-0987
(休)木曜日

名古屋
(営)AM10:00~PM7:00

ツクモ名古屋1号店
052-263-1655
(休)火曜日

ツクモ名古屋2号店
052-251-3399
(休)水曜日

札幌
(営)AM10:30~PM7:30

ツクモ札幌店
011-241-2299
(休)木曜日

(営)平日AM10:40~PM7:30日・祝AM10:10~PM7:00

DEPOツクモ2番街店
011-242-3199
(休)木曜日

★商品はお電話受け付けより、標準日数3日~1週間でお届け致します。(一部地域を除く)
★表示価格には消費税は含まれておりません。

安いのに親切
TSUKUMO
九十九電機株式会社

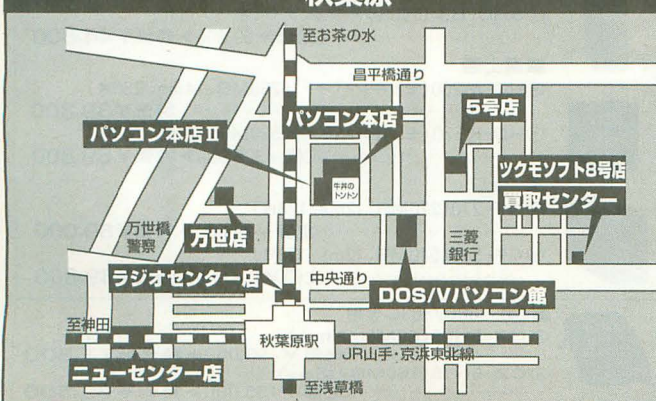
マイコン専門ショップ

P&A

SHARPエキスパートショップ

今が購入のチャンス！

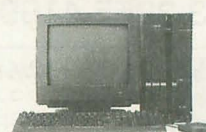
10/18〜11/17

注目!! 冬のボーナス一括払い手数料（金利）無料（平成6年11月末／12月末／平成7年1月末のいずれかをご指定下さい。）

X68030お買い得セット

（クレジット表：送料・消費税込み）

●CZ-500C
●CZ-608D(B)

定価￥492,800
P&A超特価
￥302,000

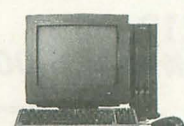
クレジット表	12回	27,400	24回	14,400
36回 10,000	48回	7,800	60回	6,500

●CZ-510C
●CZ-608D(B)

定価￥582,800
P&A超特価
￥401,000

クレジット表	12回	36,300	24回	19,100
36回 13,200	48回	10,300	60回	8,600

●CZ-300C
●CZ-608D(B)

定価￥482,800
P&A超特価
￥331,000

クレジット表	12回	30,000	24回	15,800
36回 10,900	48回	8,500	60回	7,100

●CZ-310C
●CZ-608D(B)

定価￥572,800
P&A超特価
￥396,000

クレジット表	12回	36,000	24回	18,900
36回 13,000	48回	10,200	60回	8,500

■モニター変更の場合
●CZ-615D(チューナー付)に変更の場合￥56,000
●CZ-621D(B)…………に変更の場合￥64,000
加算して下さい。

決算大処分セール 旧シリーズ今が買いどき!!

（送料￥2,000・消費税別）（クレジット表：送料・消費税込み）

X68000 Compact XVI

●CZ-674C-H
●CZ-608D(B)

定価￥392,800
P&A超特価**￥147,000**

12回	13,400	24回	7,100	36回	4,900	48回	3,800	60回	3,200

●CZ-674C-H
●CZ-608D(B)
●CZ-6FD5

定価￥492,600
P&A超特価**￥195,000**

12回	17,700	24回	9,300	36回	6,500	48回	5,000	60回	4,200

X68000 SUPER-HD（HD 80M内蔵）

●CZ-623C-TN
●CZ-608D(B)

定価￥592,800
P&A超特価**￥147,000**

X68000 PRO II

P&A特別セット 限定10台

●CZ-653C
●CZ-612D
（0.31mm、3モード TVチューナー、チルト台付）

定価￥404,800
P&A超特価**￥109,000**

X1 twin

限定10台

家庭用TV接続OK
●CZ-830C(単品)
定価￥99,800
P&A超特価
￥9,800

X68000 EXPERT II

限定10台

●CZ-603C
●CZ-604D
定価￥432,800
P&A超特価
￥98,000

X68000（単品）

限定20台

●CZ-674C(Compact XVI)‥特価￥83,000
●CZ-623C(SUPER-HD)…特価￥85,000
●CZ-653C(PRO II)………特価￥65,000
●CZ-652C(PRO)…………特価￥63,000
●CZ-602C(EXPERT)………特価￥60,000
●CZ-603C(EXPERT II)…特価￥63,000
●CZ-604C(SUPER)………特価￥75,000
●CZ-613C(EXPERT II-HD)特価￥70,000

X68000/68030用 メモリボード

（送料￥700・消費税別）

■I/Oデータ
●SH-5BE4-8M(30用)…特価￥39,500
●SH-6BE1-1ME(600C用)…特価￥10,200
●PIO-6BE1-AE(ACE/PRO/PROII用)‥特価￥10,200
●PIO-6BE2-2ME(拡張スロット用)‥特価￥21,000
●PIO-6BE4-4ME(　〃　)‥特価￥35,300

■シャープ
●CZ-5BE4(30用)………特価￥39,800
●CZ-5ME4(5BE4用増設)‥特価￥36,500
●CZ-6BE2A(XVI用)……特価￥38,900
●CZ-6BE2B(XVI、674C増設)特価￥37,500
●CZ-6BE2D(674C用)…特価￥20,500

モデム＆FAXモデム

（送料￥1,000）

〈インテグラン〉
●MP1414F(FAXモデム・ポケット型)…特価￥29,000
〈アイワ〉
●PV-AF24V5(FAXモデム・ボックス型)
……………………………特価￥21,000
●PV-AF144V5(FAXモデム・ボックス型)
……………………………特価￥29,000

〈オムロン〉
●MD-96XT10V(FAXモデム・ボックス型)……特価￥30,000
●MD-144XT10V(FAXモデム・ボックス型)……特価￥35,000
●ME1414B(FAXモデム・ボックス型)…………特価￥22,000
〈マイクロコア〉
●MC14400FX(W)(FAXモデム・ボックス型)…特価￥23,000
●MC24FC5(W)(FAXモデム・ポケット型)…特価￥20,000

MO&CD-ROM

（送料￥1,000）

■CS-M230PA(コパル)
光磁気ディスク(X68000用)
●ケーブル付
特価￥104,000

■LMO-FMX330TS
(ロジテック)●ケーブル付
定価￥168,000
特価￥97,000

■MO（ケーブル別売）
●UL-312E-S(緑電子)………特価￥62,000
●MO-120S(ICM)……………特価￥88,000
●MO-230S(　〃　)……………特価￥110,000
●LMO-340(ロジテック)………特価￥85,000
●LMO-400(　〃　)……………特価￥110,000

■CD-ROM（ケーブル別売）
⊙LU-CD2(日本テクサ) P&A特価￥15,800
⊙SCD-200(ロジテック) P&A特価￥17,000
※CD-ROM Driver＋SCSIケーブル・特価￥7,300

東京システムリサーチ製（X SIMM）

（送料￥700・消費税別）

（X SIMM VI）
⊙X VIシリーズ専用SIMM増設式メモリボード
●X SIMM VI(634C用)…定価￥16,500➡特価￥13,000
●X SIMM VIc(674C用)・定価￥16,500➡特価￥13,000
⊙増設SIMMメモリ(72PIN)
●4MB(70ns)……………………特価￥11,800
●8MB(70ns)……………………特価￥27,800
●4MB(60ns、24MHz以上用)………特価￥16,500
●8MB(60ns、24MHz以上用)………特価￥28,000

●6MB(70ns、メーカー純正品)……特価￥31,000
（X SIMM 10）⊙SIMM増設式メモリボード
●X SIMM 10………定価￥18,000➡特価￥15,700
⊙増設SIMMメモリ ●1MB×2……特価￥ 9,000
●4MB×2……特価￥30,000
●10MB例 X SIMM10＋1MB×2＋4MB×2‥￥54,700

X68000/68030専用ハードディスク

（送料￥1,000・消費税別）

外付

■ロジテック
⊙SHD-B340NU(ケーブル、ターミネータ付)
……………………定価￥59,800▶特価￥44,800

■富士通
⊙HD-M260(モッキンバード)(260MB、14ms、256K)
……………………特価￥39,800
⊙HD-K520(モッキンバード)(520MB、12ms、240K)
……………………定価￥128,000▶特価￥69,800

■ジェフ
⊙GF-270(270MB、12ms、128K)
……………………定価￥89,800▶特価￥59,000
⊙GF-540(540MB、12ms、128K)
……………………定価￥128,000▶特価￥69,800

内蔵

■CZ-500C/300C専用
⊙CZ-5H08(80MB/23ms)
……………………定価￥ 98,000▶特価￥71,800
⊙CZ-5H16(160MB/18ms)
……………………定価￥135,000▶特価￥99,500

●価格は変動します。ご注文の際は必ずお電話で価格と在庫をご確認下さい。●本広告に掲載の商品には送料及び消費税は含まれておりません。

ズバリご奉仕

《業界No.1の"P&Aメンテナンスサポート"》

最高の保証システム

①業界最長の新品パソコン5年保証
（※モニター・プリンター3年間保証.!! ※一部商品は除きます。）
②中古パソコンの1年間保証（※モニター・プリンター6ヶ月間保証.!!）
③初期不良交換期間3ヶ月（※新品商品に限らせていただきます。）
④永久買取保証
⑤配達日の指定OK!!（土曜・日曜・祭日もOK.!!）
⑥夜間配達もOK!!（※PM6:00〜PM8:00の間 ※一部地域は除きます。）

便利でお得な支払いシステム

①翌月一括払い手数料無料（ご利用下さい。）
②業界No.1の低金利!!
③月々の支払いは¥1,000より
④9ヶ月先からのスキップ払いOK!!
⑤84回までの分割、ボーナス併用OK!!
⑥カレッジクレジット
⑦ステップアップクレジット
⑧ボーナスだけで10回払いOK!!
⑨現金一括支払いOK!!
⑩商品到着払いOK!!（代引き手数料が必要になります。10万円まで900円）
（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

●法人向け
リースシステム
業務に最適なシステムを構築します。
損金処理が可能なリース契約をどうぞ。

周辺機器コーナー

（送料¥1,000・消費税別）

カラーイメージスキャナ
■JX-330X
定価¥178,000
特価~~¥128,000~~

カラーイメージジェット
■IO-735X-B
定価¥248,000
特価¥128,000

ビデオスキャナー
■CZ-6VS1
定価¥178,000
特価¥135,000

FDD（5インチ×2基）
■CZ-6FD5
定価¥99,800
P&A超特価
¥49,800

プリンター（ケーブル用紙付）
- MJ-500V2（エプソン）…特価¥44,300
- MJ-1000V2（〃）…特価¥64,300
- MJ-700V2C（〃）…特価¥78,300
- BJ-220JC（キヤノン）…特価¥58,000
- BJ-10V Lite（〃）…特価¥31,300
- BJ-15V PRO（〃）…特価¥39,700
- LBP-A404GII（〃）…特価¥99,500
- BJC-600J（〃）…特価¥78,300
- JET505J PLUS（YHP）…特価¥50,300

ペン&タブレット
■Drawing Slate
（NS CalComp）
●31090SER
定価¥74,800
特価¥58,500

- CZ-6BV1………定価¥21,000▶特価¥15,900
- CZ-8NM3………定価¥ 9,800▶特価¥ 7,200
- SH-6BF1………定価¥49,800▶特価¥36,500
- CZ-6BP1………定価¥79,800▶特価¥57,000
- CZ-6BS1………定価¥29,800▶特価¥21,500
- CZ-8NJ2（限定）…定価¥23,800▶特価¥13,800
- CZ-6CS1（674C用）・定価¥12,000▶特価¥ 8,900
- CZ-6CR1（RGBケーブル）・定価¥ 4,500▶特価¥ 3,600
- CZ6CT1（テレビコントロール）・定価¥ 5,500▶特価¥ 4,400
- CZ-6BP2………定価¥45,800▶特価¥33,300
- CZ-5MP1（X68030用）・定価¥54,800▶特価¥42,000

送料¥700・消費税別

■システムサコムボード
- SX-68MII（MIDI）
定価¥19,800
特価¥13,500
- SX-68SC（SCSI）
定価¥26,800
特価¥17,500

X68000用ソフトコーナー

（送料¥700・消費税別）

〈シャープ〉
- CZ-243BSD CYBERNOTE PRO68K・特価¥15,000
- CZ-247MSD MUSIC PRO68K（MIDI）……特価¥20,500
- CZ-249GSD CANVAS PRO68K…特価¥22,000
- CZ-257CSD Communication PRO68K Ver.2.0……特価¥15,300
- CZ-263GWD Easypaint SX-68K・特価¥ 9,800
- CZ-264GWD Easy draw SX-68K…特価¥15,300
- CZ-265HSD New Print Shop Ver.2.0……特価¥15,400
- CZ-266BSD Press Conductor PRO68K……特価¥22,000
- CZ-267BSD CHART PRO68K…特価¥29,800
- CZ-271BWD EG-Word……特価¥44,900
- CZ-272CWD Communication SX68K……特価¥14,500
- CZ-274MWD MUSIC SX68K…特価¥29,300
- CZ-275MWD SOUND SX68K…特価¥11,500
- CZ-286BSD BUSINESS PRO68K特価¥20,500
- CZ-288LWD 開発キット（work room）……特価¥29,700
- CZ-289TWD 開発キット用ツール集……特価¥ 9,600
- CZ-290TWD SX-WINDOWディスクアクセサリー集……特価¥11,500
- CZ-291BWD XOTP-SX68K……特価¥26,900
- CZ-295LSD C-Compiler PRO68K Ver.2.1 NEW KIT……特価¥32,500
- CZ-296SS/SSC SX-WINDOWS Ver.3.1……特価¥17,600
- CZ-293AW/AWC 倉庫番リベンジ SX-68K……特価¥ 5,100

〈マイクロウェア〉
- OS-9/X68030 V.2.4.5……特価¥19,900
- X-WINDOWS V.11 R5……特価¥25,500
- Technical Tool Kit V.2.4.5・特価¥17,000
- Ultra C アンド Professional Pack V.1.1……特価¥38,000
- Video PC for X680X0……特価¥57,000

〈計測技研〉
- SX広辞苑（CD-ROMバンドル版）……特価¥30,000
- Double Bookin'……特価¥ 9,600
- CD-ROM Driver V.2.0……特価¥ 3,800
- SX-Photo Gallery……特価¥13,400

〈その他〉
- F-Card V5 for X68K（クレスト）……特価¥ 9,600
- F-Calc for X68K（クレスト）・特価¥11,000
- たーみのる2（SPS）……特価¥13,000
- MU-1GS（サンワード）……特価¥21,000
- マチエール Ver.2.0（サンワード）……特価¥28,800
- Z's STAFF PRO68K Ver.3.0（ツァイト）……特価¥37,500
- Z's TRIPHONY デジタルクラフト（ツァイト）……特価¥27,000
- スーパーストリートファイターII（カプコン 5"2HD）……特価¥ 7,300

全国通販

★頭金なし!
★即日発送

●お近くの方はお立寄り下さい。専門係員が説明いたします。
●本体単品で特価で受付します。詳しくは電話にてお問合せ下さい。
●ビジネスソフト定価の20%引きOK! TELください。

P&A特選 今月の中古特選品

●CZ-674C
●68000専用モニター付
¥96,000

●CZ-623C
●68000専用モニター付
¥96,000

●CZ-653C
●68000専用モニター付
¥79,000

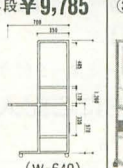
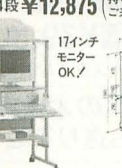
X1 twin 限定

●CZ-830C ¥9,800

- CZ-600C…¥40,000
- CZ-601C…¥40,000
- CZ-611C…¥45,000
- CZ-652C…¥50,000
- CZ-612C…¥60,000
- CZ-603C…¥53,000
- CZ-653C…¥55,000

- CZ-612C…¥65,000
- CZ-623C…¥75,000
- CZ-674C…¥73,000
- CZ-634C…¥110,000
- CZ-644C…¥145,000

※上記は単品価格、モニター別売。

高額買取り（新品もOK）格安販売

■まずはお電話下さい。
下取り専用 買取り電話▶ ☎03-3651-1884 FAX. 03-3651-0141
■下取り・買取りで、お急ぎの方は、直接当社に来店、または宅急便にてお送りください。

買取り価格…完動品・箱／マニュアル／付属品の価格です。中古販売…1年間保証付。

●下取りの場合…価格は常に変動していますので査定額を電話で確認してください。（差額は、P&A超低金利クレジットをご利用ください。）
●買取りの場合…現品が着き次第、3日以内に高価買取金額を連絡し、振込み、又は書留でお送り致します。
●近郊の方はP&A本店に直接お持ちください。即金にて¥5,000,000までお支払い致します。

●最新の在庫情報・価格はお電話にてお問い合せください。
●買い取りのみ、または、中古品どうしの交換も致します。詳しくは電話にて、お問い合せください。
●価格は変動する場合もございますので、ご注文の際には必ず在庫をご確認ください。
●本商品の掲載の商品の価格については、消費税は、含まれておりません。
●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せください。

P&A特選パソコンラック&OAチェアー（消費税込み）（送料無料、離島を除く）

①3段¥8,240 ②4段¥9,785

(W-640)

※全機種→キャスター付 ※フレーム色：ホワイト
※上から2番目棚板移動可能（4段）
※3段の場合、上から2番目の棚板は付いておりません。

③4段¥12,875（持ち帰り可能です ご来店下さい。）

17インチモニターOK!
スライドOK!

※上下2分割式/スライドマウステーブル、中棚板は2段階に可動します。
※フレーム色：グレー

①¥9,270
●布張り ダークグレー
●ガスシリンダー

②¥11,330
●布張り ダークグレー
●ガスシリンダー
●肘付

通信販売お申し込みのご案内

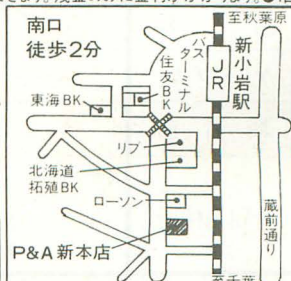
〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。●1回〜84回払いまで出来ます。但し、1回のお支払い額は¥1,000円以上。

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。（電信扱いでお振込み下さい。）

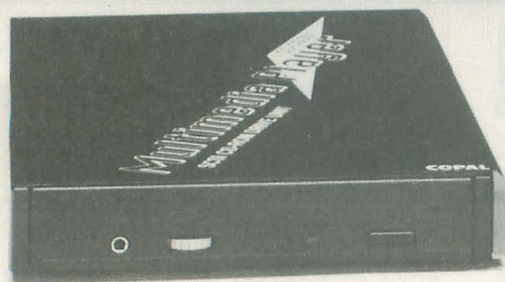
〔振込先〕さくら銀行 新小岩支店
当座預金 2408626（株）ピー・アンド・エー

超低金利クレジット率

回数	3	6	10	12	15	24	36	48	60	72
手数料	2.6	3.5	4.4	4.9	7.8	10.4	14.4	18.9	24.4	31.8

※お支払いは、便利な商品到着払い〈手数料（10万円まで900円）要〉をご利用下さい。

株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目2番地20号
●営業時間：AM10:00〜PM7:00 日・祭：AM10:00〜PM6:00
☎03-3651-0148（代）
●定休日／毎週水曜日 FAX.03-3651-0141 MAC/DOS Vフロア ☎03-3655-4454

■姉妹店／ユーアイネット ギガ店OPEN（千葉県浦安市入船1-4-1ショッパーズ1F ☎0473-81-2107）もよろしく!!

X680x0にジャストフィット
精悍な黒モデル フルラインナップ

エアフィルタ交換不要の3.5インチ光磁気ディスクユニット
CS-M120PX
定価¥178,000 **通販特価¥89,800**

■平均シークタイム30ms,回転数3600rpm,記憶容量128MBの高性能ドライブ。
■今回お買い求めの方に限りケーブル・ターミネーターをサービス。
＊X68000,Human68Kでのご使用となります。SX-WINDOWでのご使用についてはお問いあわせください。

外付ハードディスクユニット
CS-H540X
定価¥128,000 **通販特価¥59,800**

■フォーマット容量540MB,平均アクセスタイム12ms,ターミネータ付,ケーブルはサービス

CS-H240X
定価¥79,800 **通販特価¥39,800**

■フォーマット容量240MB,平均アクセスタイム15ms,ターミネータ付,ケーブルはサービス

デバイスドライバー付倍速CD-ROMユニット
CS-CD301X
定価¥59.800 **通販特価¥29,800**

■各種フォーマット対応 CD-DA,XA,Photo-CD,CD-Bridge,CD-1フォーマット対応
■キャディのいらないトレー式，ケーブル/ターミネータ標準添付(ディジーチェーン接続が可能)

＊4機種ともSCSI I/Fボードはパソコン本体に附属のものまたは純正品が使用可能です。その他サードパーティ製のSCSI I/Fボードとの接続についてはお問い合わせください。
＊ご注文の際にはご希望のケーブルをご指定下さい。
(CS-H540X、CS-H240Xについては、ユニット側はフルピッチコネクタで、その他の機種はハーフピッチコネクタです。)

●製品についての情報は、FAXステーションから次の要領で取り出して下さい。

FAX Station

1. FAXの受話器をあげて
2. FAXステーション(☎03-3499-0177)にダイヤルして下さい。
3. 音声案内に従って(ダイヤル回線の方はピポパのトーン信号に切り換えて)#を押します。
4. 音声案内に従って情報番号6200#を押し、最後に終了の#を押します。
5. 送受信のメッセージ終了後(約3秒後ピー音を確認)ファクシミリのスタートボタンを押して受話器を戻します。→「製品情報」をお受取下さい。

●お申し込みは、注文書の太枠線内にご記入の上FAXまたは郵送にてお送り下さい。

●お申し込み先　コパル綜合サービス株式会社　通販係
〒174　東京都板橋区志村2-16-20
TEL.03-3965-1144　FAX.03-3968-1029

*商品の技術的なご質問·ご相談はユーザーサポート係まで
TEL.03-3965-1161

●お支払いは銀行振込で、下記口座までお振込下さい。
(振込手数料はお客様負担で電信扱いでお振込下さい)

口座番号　第一勧業銀行　志村支店　普通預金　No.1369382
口座名義　コパル綜合サービス株式会社

●商品の引渡しは代金お支払い後となります。
●商品はご入金後、原則として3日以内に発送します。
(在庫切れの場合は、ご連絡いたします。)

■ご注文書
FAX 03-3968-1029

品名		ご注文台数　台	ご連絡先 TEL.　(　) FAX.　(　)
ケーブル※1	□フル～ハーフ □ハーフ～ハーフ □フル～フル		
お名前	ふりがな		
お届先住所	(〒　-　) 都道府県　区市郡		1.会社　2.自宅

弊社記入欄	
受付番号	
受付日	
納入日	
備考	

※1 ご希望のケーブルをご指定ください。

好評発売中

X68k Programming Serise #3

X680x0 TEX

吉野智興・川本琢二・山崎岳志・実森仁志　共著

●B5変形判・2冊組・ビニール箱入り・5"FD8枚組　定価9,800円

『Vol.1 User's Guide編』では、はじめてTeXを使う人のために簡単なインストーラによるTeXの基本的な使い方の解説を、すでにTeXを使い込んでいる人のためにはカスタマイズのしかたや、数学記号などの表記に優れたAmSTeX、楽譜が書けるMUSIC-TeXなどのサンプルや、縦書きマクロ(アスキー、インプレス開発)などの周辺ツールの解説をしています。

また、『Vol.2 Reference編』ではTeX、METAFONT、fontman、preview、print、makefontなどの、環境変数、オプションなどの解説をまとめてあります。

X68k Programming Series　追補版と改訂版　3冊同時発売中

X68k Programming Series ##

X680x0 Develop & libc II

吉野智興・中村祐一・石丸敏弘・今野幸義・村上敬一郎・大西恵司　共著

●B5変形判・5"FD2枚組●定価2,900円

「X68k Programming Serise #1 X68000 Develop」収録のGCC、HAS、HLK、GDBと「X68k Programming Serise #2 X680x0 libc」収録のライブラリをX68030でも動作するようにバージョンアップした追補版です。

バージョンアップによって変更あるいは追加された機能と、約1年に渡るバグ報告を元に修正された機能について解説します。

付属FDには、最新のプログラムを収録しました。

X68k Programming Series #1

X680x0 Develop Manual Book

吉野智興・中村祐一・石丸敏弘・今野幸義　共著　B5変形版・2冊組・箱入り●定価5,000円

X68k Programming Series #2

X680x0 libc Manual Book

村上敬一郎・大西恵司・荻野祐二　共著　B5変形版・2冊組・箱入り●定価6,300円

それぞれ前作のマニュアル部分をまとめた改訂版です。

「X680x0 Develop & libc II」を発行するにあたり、変更・修正された機能についても解説しています。

SOFT BANK　ソフトバンク株式会社/出版事業部　〒103 東京都中央区日本橋浜町3-42-3 TEL.03-5642-8100　定価は税込です。

超級特価

X68030(CZ-500C)+040turbo/バンドルセット ¥328,000!

オリジナルアプリケーション開発速報

1994年11月1日発行（毎月1回1日発行）通巻14号 *FirstClassTechnology* 発行

X680x0用Ether net接続パック

EthernetStarterPack/X680x0

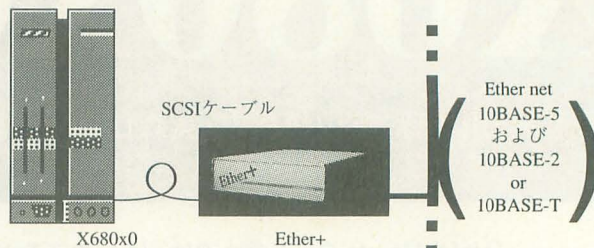
発売記念特価 ¥78,000

ESP/Xは、Ether netアダプタ「Ether+」と、TCP／IPドライバ、そして基本的なアプリケーションからなるパッケージです。

・Ether+(米コンパチブルシステムズ社製)
SCSIインターフェースを介してEther netとX680x0を接続するためのハードウェアです。
※10BASE-2対応モデル・10BASE-T対応モデルの2種類があります。

・TCP/IPドライバ
X680x0でTCP/IPをサポートするドライバ。ソケットも利用可能です。

・基本的なアプリケーション
ftp、telnet(いずれもクライアント)等、基本的なアプリケーションを標準添付。ドライバを活用するためのライブラリも付属します。

SCSIケーブル
X680x0
Ether+
Ether net 10BASE-5 および 10BASE-2 or 10BASE-T

簡単な接続で、X680x0がワークステーションの間で働き始めます。
※NetWareには未対応です。

●動作環境
・Human68k ver3.0以上
・メモリ常駐量500KB前後
・SCSIインターフェース内蔵機種以外はSCSIボードが必要

SX-WINDOW用CD-ROM辞書検索ソフト

SX広辞苑

《EPWING対応版》

標準価格 ¥19,800
岩波書店「広辞苑第4版CD-ROM版」バンドルセット ¥43,800

SX広辞苑《EPWING対応版》は、「広辞苑第4版CD-ROM版」を効率的に検索し、120%活用するためのソフトです。

●SX広辞苑《EPWING対応版》の特長
・豊富でパワフルな検索方法により、必要な情報をすばやくピックアップ。
・使う側に立って操作系をリニューアル。さらに簡単に、さらに鋭く作業を行なえます。
・広辞苑の最新版である第4版をもとにしたCD-ROMを使用するので、よりコンテンポラリーなキーワードにアクセス可能です。
・SX-WINDOW上で動作するので記事の参照や引用がとても簡単。シャーペンやEGWordと組み合わせて活用できます。（ただし、広辞苑では大量の引用は禁止されています）
・シャーペンと融合して語句の検索を行なうシャーペン用外部コマンド"LightWing.X"を同梱。複雑な検索を行なう場合はSX広辞苑.Xを、普段よく使う単純な検索にはLightWing.Xを、という使い分けも可能です。
・広辞苑第4版CD-ROM版と同様に、EPWING(V1)規約にもとづいたCD-ROMタイトルなら、ほとんどのCD-ROMの内容を検索できます。

●動作環境
・SX-WINDOW 3.0以上
・SX-WINDOW動作中の空きメモリとして1MB以上を推奨
・CD-ROMドライブ(CD-ROM Driver Ver2.0が付属するので、CD-ROM Driverを別途お買い上げいただく必要はありません。CD-ROM Driverのマニュアルや添付ソフト等は付属しません)

X68030用 68040搭載アクセラレータ

X68040turbo

標準価格 ¥98,000 ヒートシンク別売 ¥1,000

040turboは、68040を搭載したX68030(5インチタイプ)専用のアクセラレータです。040turboを装着することで得られるパフォーマンスは、従来の2～3倍! 計算、特に浮動小数点演算中心のソフトならば、さらにそれ以上の高速化も望めます。

詳しくはソフトバンク刊「X68040turbo～A Story of Making "After X68030"～」(BEEPs著)をご覧ください。

040turboは当社のショップBASIC-HOUSEでの直販、および通販でのみお買い求めいただけます。ご注文いただいてからしばらくお待ちいただく場合もありますので、お早めにご注文ください。

バージョンアップのお知らせ!!

040Turboに一部不具合がありましたので、バージョンアップサービスをいたします。
症状:040モードでごくまれにバスエラーの処理でハングアップする場合がありました。
対象となる方:IC2に「V5」と印刷されている040Turboをお使いの方。
バージョンアップ方法：80円切手を貼り、住所氏名を記入した返信用封筒を同封の上、基板からIC2を抜き、封書でお送りください。IC2の不具合を修正後、お送りいたします。
GALの梱包には、足が折れないよう、段ボールに挿す等の対処をお願いいたします。
所要期間は1週間程度とお考えください。
送付先:〒320 栃木県宇都宮市京町11-18 OYAMAビル2F
FirstClassTechnology 040turbo係

バージョンアップ!!

CD-ROM Driver Ver2.00

標準価格 ¥4,800

CD-ROM DriverがバージョンアップしてVer2.00になりました。
SCSI-2コマンド対応CD-ROMドライブ専用となって、より安定して動作します。当社サポートネットTECOSYS-3でVer1.06からのバイナリ差分をダウンロードしていただけます。
※SCSI-2対応ドライブ以外をお使いの方はバージョンアップの必要はありません。

SX-WINDOW用スケジュール管理ソフト

DoubleBookin'

標準価格 ¥12,800

発売中

SX-WINDOW用Photo-CDビュアー

SX-PhotoGallery

標準価格 ¥15,800
通販特価 ¥15,000

X680x0用フリーソフトウェア集CD-ROM

FreeSoftwareSelection Vol.2

標準価格 ¥6,000

お求めはお近くのパソコンショップ、または当社通販部(TEL:0286-22-9811)へお申し込みください。
通販ご希望の方は、ソフト代金＋送料1,000に消費税を加え、ご住所・お名前・電話番号・商品名を明記した紙を同封の上、現金封筒でお申し込みください。

サポートネット TECOSYS-3稼動中!

アクセス番号:(0286)51-1430 (9600bpsMNP5)

計測技研の製品の最新情報がわかっちゃう、とっても嬉しいネット、TECOSYS-3。24時間稼動中です。ソフトやハードのQ&Aやバージョンアップ情報などなど、68ユーザーなら見逃せない情報が満載...の予定(^_^; Nifty等、一般的なパソコンネットと同じ設定でアクセスできるから、初めてでも安心(^^) アクセスお待ちしております。

低金利クレジット 通信販売送料 全国一律¥1,000 長期クレジット可能
※表示価格に消費税は含まれておりません

株式会社 計測技研 マイコンショップBASIC HOUSE
本社／ショールーム／通販部
〒321 栃木県宇都宮市竹林町503-1
TEL 0286-22-9811 FAX 0286-25-3970

※ 記載されている会社名および商品名は各社の登録商標もしくは商標です。

アーケード版ではまだ体験できない新しい上海

ELECTRONIC ARTS

上海 万里の長城

Shanghai : THE GREAT WALL

めざせ、長城踏破。

積み重ねられた麻雀牌の中から、同じ種類のものを取り除いていくパズルゲーム、上海。
パズルゲームの傑作、あの上海がさらにグレードアップして、登場します。
秦の始皇帝が築きあげたといわれる歴史的建造物、万里の長城を舞台に、あなたの思考力を試してください。

- 取り除く条件がそろった２つの牌を探し出し、制限時間内にすべての牌を消すとゲームクリアになります。
- クラシック上海、グレートウォール（牌落下型上海）、ディムサム（牌スライド型上海）、バトル上海（対戦型上海）の４つのモードが用意されています。
- プレイヤーは万里の長城の始点（山海関）から終点（嘉峪関）まで、ゲームをクリアしながら進んでいきます。
- ２人協力プレイも可能です。

クラシック上海
壁になった牌を消していきます。下の牌が取り去られると、上の牌が落下します。落下した牌は右や左に他の牌がある場合、その牌に吸い寄せられます。

ディムサム
同種類の麻雀牌を選び消していく、従来型の上海です。

グレートウォール
牌をスライドさせて消していきます。

バトル上海
対戦型上海です。

© ACTVISION

11月発売予定

PC-9801シリーズ（5インチ）
価格：8,800円　メディア：5"FD

PC-9801シリーズ（3.5インチ）
価格：8,800円　メディア：3.5"FD

X68000シリーズ
価格：8,800円　メディア：5"FD

3DO版
近日発売

FM-TOWNS
価格：8,800円　メディア：CD-ROM
12月発売予定

好評発売中

シンジケート
PC-98 FD版（5", 3.5"）
PC-98 CD-ROM版
FM-TOWNS版,
DOS/V版,
MAC版
各12,800円
©1993/1994 Bullfrog Production Ltd.

ストライクコマンダー

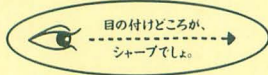
PC-98 FD版（5", 3.5"）
PC-98 CD-ROM版
FM-TOWNS版,
DOS/V版
各12,800円
©1992/1993 ORIGIN Systems, Ltd.
Strikecommander is a trademark of ORIGIN Systems,Inc. Origin is a trademark of ORIGIN Systems,Inc.
©1994 Electronic Arts.

シャドウキャスター

PC-98 FD版（5", 3.5"）
PC-98 CD-ROM版
各12,800円
© 1993 Raven Software,Inc.
© 1993/1994 Electronic Arts.
Shadowcaster is a trademark of ORIGIN Systems,Inc.

エレクトロニック・アーツ・ビクター株式会社

●画面は開発中のものです。●詳細な動作環境などにつきましては、パッケージの裏面、もしくは当社カスタマーサポート係まで電話にてお問い合わせください。●製品および社名は各社の商標および登録商標です。●価格は税別価格です。

お問い合わせ先　カスタマーサポート係　〒150東京都渋谷区神宮前2-4-12フルークス外苑　TEL.03-5410-3100　13:00～16:00・月～金（祝日休）

感性を光らせる。

さまざまなフィールドで、研ぎ澄まされた感性に応える潜在能力の実証

X68の潜在能力は、まさに時代とともに証明されつつあります。
開発当初より、現在のマルチメディア環境を想定していた事実。
グラフィック能力はもちろん、ADPCM対応、オリジナルウィンドウシステム、
X68にとってこれらは、数年前のスペックなのです。
パソコンの存在そのものを革新した「創造性」、マインドを喚起する「こだわり」、
いま、先見のユーザーに支えられたX68は
そのコンセプトの開花を得て、多彩なフィールドへと飛翔します。

Workbench

WSとしての楽しみ

たとえば、リアルタイム・マルチタスク・オペレーティング・システムOS/9。
X68030の能力を最大限に引き出す
UNIXライクな操作性と洗練された機能。
X-WINDOWや動画ツールのサポートで
さらに深い楽しみが…。

※OS/9はマイクロウェア・システムズ㈱の登録商標です。
※UNIXは、X/Openカンパニーリミテッドが独占的にライセンスする米国および他の国における登録商標です。

Create

創造するよろこび

SX-WINDOW開発支援ツールが
創造力を刺激する。
ソフト開発に必要なツールや
サンプルプログラムを多彩にバンドル、
ウィンドウ上で効率よく作業でき、
初めてプログラムに挑む人への
やさしい配慮が、創造するよろこびを
さらに高めてくれるでしょう。

Ammusement

遊びへのこだわり

X68の能力の高さを端的に示す
アミューズメントフィールド。
マインドをきわめたゲームフリークの
熱い期待に応える。
画像の美しさが感性を刺激する、
さらにパワーアップされた
「スーパーストリートファイターII」なら、
キミのこだわり度は今、全開！

© CAPCOM ALL RIGHTS RESERVED

X68030 32bit PERSONAL WORKSTATION / X68000 PERSONAL WORKSTATION ·XVI

X68030 [本体+キーボード+マウス・トラックボール]
130㎜FD(5.25型)タイプ CZ-500C-B(チタンブラック) 標準価格398,000円(税別)・〈HD内蔵〉CZ-510C-B(チタンブラック) 標準価格488,000円(税別)

X68030 Compact [本体+キーボード+マウス]
90㎜FD(3.5型)タイプ CZ-300C-B(チタンブラック) 標準価格388,000円(税別)

X68000 XVI Compact [本体+キーボード+マウス]
90㎜FD(3.5型)タイプ CZ-674C-H(グレー) 標準価格298,000円(税別)

●ディスプレイは別売です。●消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は、標準価格には含まれておりません。●画面はハメコミ合成です。

■お問い合わせは… シャープ株式会社 電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

T1002179110688 雑誌 02179-11